REDUCTION OF SEQUENTIAL SWITCHING

SYSTEMS TO MINIMUM FORMS

A THESIS

Presented to

The Faculty of the Graduate Division

by

Clifford O. Guffee, Jr.

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

Georgia Institute of Technology

April, 1964

REDUCTION OF SEQUENTIAL SWITCHING

SYSTEMS TO MINIMUM FORMS

Approved:

Dr. D. C. Fielder, Chairman

Dr. W. B. Jones, Jr.

Dr. Wm. F. Atchison

Date approved by Chairman: May 30, 1964

ACKNOWLEDGMENTS

The author wishes to thank his thesis advisor, Dr. D. C. Fielder, for continued guidance and encouragement during the course of the investigation. Special thanks are also due the other members of the reading committee, Dr. W. B. Jones, Jr., and Dr. W. F. Atchison, for reviewing the manuscript and providing constructive criticisms. Appreciation is also expressed to the staff of the Rich Electronic Computer Center for their advice and suggestions in the preparation of the computer program.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SUMMARY

The purpose of this dissertation is to develop a unique technique for the systematic reduction of a sequential switching system to minimum form. The restrictions upon the technique are that it must always result in a complete reduction and that it may easily be programmed for a computer.

A method has been developed herein which will always allow the determination of the minimum flow matrix description of a sequential switching circuit which is possible through merging. Rules and procedures are established which allow the procedure to be rendered completely automatic. The procedure is based upon the work of two authors; however, the application of these contributions differ from that described by either author.

Huffman presents the initial method for the systhesis of a sequential switching circuit (3). The procedure presented herein relies upon Huffman's concepts of redundant states and of merging. Another author, Ginsburg, considered the reduction of sequential circuits from the concept of compatible states (11). Ginsburg also presents a method for determining the size of the theoretical minimum attainable flow matrix.

The concept of compatible states is extended so as to be able to determine the compatible states of Huffman's flow matrix. This then allows the determination of the size of the theoretical minimum flow matrix of Huffman's notation. The theoretical minimum size gives a lower

bound for the reduction procedure, and is used as a reference in the automatic reduction of a flow matrix.

The reduction procedure herein introduces the concepts of merger-restriction tables and restriction-dependency tables. Compatible state-pairs are determined for the flow matrix under consideration, and rules are presented which allow the determination of the restriction(s) which may exist upon a possible merger. A listing of the possible mergers and their restrictions then constitutes the merger-restriction table. It is shown herein that the restrictions will be a subset of the set of redundant states of the flow matrix; however, it is also shown that not all redundant states must be considered as a restriction when reducing a flow matrix. The rules, which are given for establishing the merger-restriction table, will automatically consider only those redundant states which must be considered as such. It is shown that the basic tables may be established for any flow matrix which represents a sequential circuit action; hence, the automatic reduction is independent of the flow matrix notation that may be used.

Dependencies among the restrictions are also discussed, and a restriction-dependency table is established. Based upon this table, rules are presented which allow a systematic and complete examination of the possible combinations which exist among the restrictions. The examination procedure may easily be programmed for an automatic reduction procedure.

The total procedure involved in reducing a flow matrix is then discussed, and rules are presented which allow the total reduction to be done automatically. The rules and the reduction procedure are based upon use of the established tables, and a chapter is devoted to discussing a computer program that has been written so as to incorporate all the basic ideals of the reduction procedure.

In order to render the procedure completely automatic, it was necessary to develop some optimum merging rules. These rules are applied to a merger diagram, as established by Huffman (3), in order to determine the resultant minimum state flow matrix. The rules are based on the premise that certain mergers may be removed from the merger diagram and combined in the reduced flow matrix; then, if the remainder of the merger diagram is reduced in an optimum manner, the minimum state flow matrix will always be obtained.

Examples are presented in order to illustrate the different types of dependencies which exist among the states of a flow matrix and to clearly illustrate the reduction procedure. The rules and tables presented herein will allow the complete reduction of a flow matrix in a systematic manner through merging, despite the size of the flow matrix or the enormity of the dependencies which may exist among the states of the flow matrix.

CHAPTER I

INTRODUCTION

The design of sequential circuits is generally conceded to be more difficult than the design of combinational switching circuits. In the latter design problem, comparatively simple and straightforward methods of synthesis have been known for some time (1), (2). However, prior to 1954, no practical method for the synthesis of sequential switching circuits was known.

In 1954, Huffman introduced a method for the complete design of a sequential circuit (3). Huffman's procedure utilized a "flow matrix" which describes the sequential action of the circuit in terms of input and output states. By "secondary assignment" methods, the sequential problem can be reduced to the more easily solved combinational circuit problem.

A fundamental step in Huffman's technique is the reduction of the rows of the initial flow matrix in order to achieve the most economical circuit. This process is one of row reduction and is called "merging." Huffman introduced several graphical and numerical aids to facilitate the merging process.

Mealy considered state diagrams as an aid to examining sequential truth tables (6) and investigated the simplest means of completely reducing a flow matrix. In his published work, Mealy included a note on the problems of synthesizing relatively large circuits. He questioned the possibility of doing this with any method that relies upon truth tables without making use of automatic design procedures inasmuch as large truth tables become unmanageable.

Several authors have considered the automatic synthesis of sequential circuits. Aufenkemp and Hohn considered the reduction of the flow matrices which describe a sequential circuit (7) and Aufenkemp extended the method in a later article (8).

Other authors have considered the assignment of internal codes to the internal states of the flow matrix. One such article was written by Armstrong (16) in which he describes a method which was successfully programmed.

All reduction methods for sequential circuits considered so far rely in part upon the well-known merging techniques (8), (3), (5). However another author, Ginsburg, has described an example of a flow table which cannot be minimized through the use of the classical merging procedures (11). In his article, Ginsburg outlines his procedure and noted that although the execution of part of his procedure was automatic and could easily be programmed on a computer, the basis of his method depended on the enumeration and the selection of numerous alternative paths. At present, no rules exist which would allow the procedure to be rendered completely automatic.

This dissertation is concerned with the automatic reduction of flow matrices. To do this, it is first necessary to examine Huffman's merging procedures (3), (5), and to partially examine the procedure presented by Ginsburg (11). With this as a basis, a reduction procedure is then developed which may be used to reduce any flow matrix by merging. The procedure may be used to automatically reduce a flow matrix and will exhaustively examine even the most complicated flow matrix. The procedure may be applied equally well to either Huffman's or Ginsburg's flow matrix. The example presented by Ginsburg is analyzed by the procedure, and where the procedure fails to obtain the minimum state flow matrix, a double merging process is considered as a possibility of allowing Ginsburg's procedure to be rendered automatic.

Finally, the established procedure of this dissertation will be compared to the automatic procedures of Aufenkemp and Hohn (7), (8). Their examples will be analyzed by the automatic procedure presented herein, and the results will be compared to the results presented in their papers.

CHAPTER II

HUFFMAN'S SYSTHESIS PROCEDURE

Outline of Synthesis Procedure

The following synthesis procedure was developed by D. A. Huffman and is explained in several articles (3), (5). S. H. Caldwell, in his text (5), explains the procedure as follows:

(1) From a description of the problem given by word statement, write a primitive flow table which by definition includes a listing of the output associated with each stable state.

(2) Verify that the primary flow table correctly describes the circuit performance for all possible input sequences.

(3) Prepare a merger diagram and from it decide how the rows of the primitive flow table are to be merged in the final flow table.

(4) Write the merged flow table.

(5) Prepare a transition diagram and from it assign the row states.

(6) Write the excitation matrix.

(7) Write the output matrix.

(8) Write the excitation and output functions from their respective matrices and design the combinational circuit thus specified.

The procedure, thus defined, reduces the design of sequential switching circuits to the more easily designed combination circuit.

The procedure, thus defined, reduces the design of sequential switching circuits to the more easily designed combinational circuit.

It will be assumed herein that the primitive flow matrix has been specified and verified (steps 1 and 2). Also, the assignment of secondary row states (step 5) and the design of the circuit (step 8) will follow at a later time. The steps which are of interest herein are then steps 3, 4, 6, 7. Taken together, these four steps result in reducing the number of components required in the final circuit and any economizing in design must first be obtained through the application of these steps.

Huffman's Flow Matrix

It is necessary to examine the primitive flow matrix (hereafter referred to as the flow matrix) to understand how it describes the sequential circuit action. A typical flow matrix is shown in Figure 1; all flow matrices of Huffman's type will have the same form.

X_1	X_2	X_3	X_4	Z_1
1	3	-	6	01
-	3	2	6	10
1	3	5	-	01
4	3	-	6	10
-	3	5	6	01
4	-	2	6	10

Figure 1. Illustration of Huffman's Flow Matrix

The allowable inputs of a sequential circuit are shown as X_i, and the allowable outputs as Z_i*. Every sequential circuit is assumed to have a finite number of different inputs, a finite number of different outputs, and a finite number of different internal states. In the flow matrix, each internal state is assigned an arbitrary integer as an identifier; however, after the assignment, the internal state and the integer are uniquely associated together.

It is possible to distinguish between internal states by classifying them as either stable or unstable internal states. A stable internal state exists whenever the sequential circuit is in a quiescent condition for a fixed input. However, the number of different internal stable states does not have to equal the number of different inputs. Unstable internal states are those internal states a sequential circuit assumes as a transition is made from one internal stable state to a second internal stable state because of a change in the input.**

The transition between stable states is controlled entirely within the sequential circuit, but is uniquely determined by the previous stable state and the change in the input; hence, all unstable states are uniquely associated with a stable state. That is, in the flow matrix representation, each different stable state is assigned a unique integer, and any unstable state through which the circuit must pass in attaining a specified

---

*In the reduction of a flow matrix, knowledge of the actual outputs of the circuit is unimportant as long as it is known which outputs are different. Thus, Figure 1 represents all outputs together as a single binary number.

**Hereafter, internal stable states and internal unstable states will be noted as stable states and unstable states respectively.

stable state is assigned the same integer identifier as that held by the stable state. In the flow matrix, a stable state is enclosed within a circle to distinguish it from an unstable state.

An example of the flow matrix representation of a sequential circuit can now be described with the aid of Figure 1. If the sequential circuit should be in a stable state (2), the output would be $Z_1 = 10$. At this time, the previous input is known to be X_3. If input X_2 should now occur, the circuit would advance to unstable state 3, and through internal control advance to stable state (3) with the new output being $Z_1 = 01$.

If any row of the flow matrix contains a blank entry corresponding to a particular input, then this specifies that the particular input will not occur while the circuit is in a stable state assigned to that row. More than one stable state may be assigned to any row as long as each stable state has the same required output.

Secondary State Assignment

The assignment of secondary states is the process of allotting internal memory elements to the stable states of the switching circuit. Each memory element can distinguish between no more than two states, and S_o, the minimum number of memory elements required for a sequential circuit, is specified by the inequality

$$2^{S_o} \geq M , \qquad (1)$$

where M is equal to the total number of rows in the flow matrix.

It is necessary that S_o be an integer; therefore, a matrix with three rows will require a minimum of two memory elements (the same as required for a matrix with four rows). Switching hazards (5) will sometimes cause an increase in the minimum number S_o; however, this eventuality is resolved in the assignment of secondary states.

From the constraint on S_o, it is seen that the first step in reducing a sequential circuit must be the reduction of the number of rows in its flow matrix representation.*

Huffman's Reduction Procedure

In general, there are two distinct methods of row reduction included in Huffman's procedure, and in any problem each procedure is applied separately. Listed in order of application, the procedures are:

(1) Removal of the redundant states [step (3)].

(2) Merging of the states [step (3)], and the forming of the excitation and output matrices [steps (5) and (6)].

Reduction methods (1) and (2) are distinguished by their application, although they are not independent in the reduction of a matrix. The following paragraphs will explain each of the above procedures.

---

*In order to be consistent with flow matrix notation as used by other authors (as will be introduced in later chapters), it is necessary to refer to the rows of Huffman's flow matrix as states. The state of a flow matrix should not be confused with specified states such as internal state, stable state, or unstable state as referred to earlier by circled and uncircled numerical entries. Thus, the flow matrix of Figure 1 is a six state matrix (i.e., six rows), and state 3 would be the third row, and is not to be confused with stable state 3 or unstable state 3.

Redundant States

A redundant state results when two or more stable states within the same input column produce the same output, and any sequence of allowable inputs beginning at either stable state produces the same sequence of outputs. From the above definition, it is seen that when redundant states occur, one state may fulfill the function of any or all of the redundant states.

An example of a flow matrix which contains redundant states is shown in Figure 2(a). In this example, states 1 and 2 are redundant since 1 and 2 occur within the same column, have the same output, and any sequence of inputs will advance either stable state to the same ultimate stable state; i.e., input X_2 will advance either 1 or 2 to 3 , while input X_4 will advance either 1 or 2 to 7 . Input X_3 is not specified and, therefore, will not occur while the circuit is in stable states 1 or 2 .

An important use of the unspecified input is noted in conjunction with 5 and 6 . It is seen that these two internal states will allow the two states which contain them to be redundant except possibly for input X_1. Input X_1 will advance 6 to 1 , while 5 is unspecified. However, it is permissible to overspecify* state 5 and require that it is also advanced to 1 for input X_1. With this advantage, states 5 and 6 are then redundant and may be combined as one state.

*Overspecifying the flow matrix (5) will not affect the original sequential circuit requirements as the flow matrix was originally specified for every allowable sequence of inputs, and hence the physical circuit will never attain any internal state which is now added in place of a blank entry.

The two sets of redundant states have been combined, and the reduced flow matrix is shown in Figure 2(b). The changes, from Figure 2(a) to Figure 2(b), are the removal of states 2 and 6 and the replacement of unstable state numbers 2 and 6, in the remaining flow matrix, by unstable state numbers 1 and 5, respectively.

Suppose the flow matrix of Figure 2(a) is respecified by replacing the blank entry in column X_1 by unstable state number 2. This is shown in Figure 2(c), and it allows input X_1 to occur while the circuit is in State 5. All of the requirements for redundancy of states 5 and 6 are now satisfied except for input X_1. It is now necessary to show that states 1 and 2 are equivalent before states 5 and 6 may be redundant. This has been shown; therefore, the respecified flow matrix will reduce to Figure 2(b) as before.

X_1	X_2	X_3	X_4	Z_1
1	3	-	7	0
2	3	-	7	0
1	3	6	-	1
2	4	5	-	0
-	4	5	8	0
1	4	6	8	0
2	-	5	7	1
1	-	6	8	0

Figure 2(a). A Flow Matrix Illustrating Redundant States

X_1	X_2	X_3	X_4	Z_1
1	3	-	7	0
1	3	5	-	1
1	4	5	-	0
1	4	5	8	0
1	-	5	7	1
1	-	5	8	0

Figure 2(b). Reduced Flow Matrix for Figure 2(a) After Removal of Redundant States

X_1	X_2	X_3	X_4	Z_1
2	4	5	8	0
1	4	6	8	0

Figure 2(c). Modification to Flow Matrix of Figure 2(a) to Illustrate a Dependency Among Redundant States

Consider now an example of a flow matrix in which the redundant states are not so obvious. At first inspection of Figure 3(a), there are several states which almost fulfill the conditions of redundancy. These are states 2 and 6, states 3 and 10, and states 4 and 11. Each of these pairs of rows must be examined more closely with respect to a single input.

X_1	X_2	X_3	X_4	Z_1
1	7	9	4	11
2	5	3	4	01
1	7	3	11	10
2	-	3	4	00
6	5	9	-	11
6	5	3	11	01
1	7	14	-	10
1	7	9	13	01
1	7	10	4	10
2	5	10	11	00
2	-	14	13	11
2	5	14	11	00

Figure 3(a). A Flow Matrix Illustrating Dependency Among the Redundant States

X_1	X_2	X_3	X_4	Z_1
1	7	9	4	11
2	5	3	4	01
1	7	3	4	10
2	5	3	4	00
2	5	9	-	11
1	7	14	-	10
1	7	9	13	01
1	-	14	13	11
2	5	14	4	00

Figure 3(b). Reduced Flow Matrix of Figure 3(a)

After a closer examination it is seen that states 3 and 10 are redundant only if states 4 and 11 are redundant; also, states 4 and 11 are redundant only if states 3 and 10 are redundant. It is possible to conclude that both pairs are redundant; hence, each pair may be replaced by a single state. The resultant flow matrix is shown in Figure 3(b).

Figures 3(c) and 3(d) show the excitation and output matrices for this example. Note that a further reduction has been obtained, and the reduced matrix now contains only eight states. The use of the excitation and output matrix is explained in the next section.

X_1	X_2	X_3	X_4
1	7	9	4
2	5	3	4
1	7	3	4
2	5	9	-
1	7	14	-
1	7	9	13
2	-	14	13
2	5	14	4

Figure 3(c). Output Matrix for the Example of Figure 3(a)

X_1	X_2	X_3	X_4
11	-	-	-
01	-	-	00
-	-	10	-
-	11	-	-
-	10	-	-
-	-	01	-
-	-	-	11
-	-	00	-

Figure 3(d). Excitation Matrix for the Example of Figure 3(a)

Figure 4 is a flow matrix described by Caldwell (5). For this flow matrix, the dependency among the redundant states is even more complicated, but an exhaustive analysis shows that the flow matrix reduces to that of Figure 3(b).

Caldwell (5) gives an "equivalent relations diagram" for the study of the dependency which may exist among the redundant states. However, a much more systematic method is given by Marcus (14). Marcus establishes a "tabular approach" which could easily be programmed for use in a digital computer flow matrix reduction procedure.

X_1	X_2	X_3	X_4	Z_1
1	7	9	4	11
2	5	3	4	01
1	7	3	11	10
2	-	3	4	00
6	5	4	-	11
6	5	3	11	01
1	7	14	-	10
8	12	3	4	01
1	7	9	13	01
1	7	10	4	10
8	-	10	11	00
6	12	9	-	11
8	-	14	13	11
2	12	14	11	00

Figure 4. A Flow Matrix Illustrating Dependency Among the Redundant States

The dependency among redundant states will be considered in more detail in a later chapter.

<u>Mergers and the Excitation and Output Matrices</u>

The number of states in a flow matrix may also be reduced by a process of "merging." Merging of states (5), (3), is normally considered after the redundant states have been removed; however, merging is a self-contained process and may be applied at any time to a flow matrix which is being reduced.

Merging of states is possible whenever one state may fulfill the function of two or more states. It is distinguished from removing redundant states by the fact the stable states which are being merged do not contain circled entries within the same input column of the flow matrix.

Two states may be merged if for every input X_i:

(1) both states contain the same unstable state number, or

(2) one state contains a stable state number while the second state contains the same number as an unstable state, or

(3) one or both states contain a blank entry.

If more than two states are to be merged as a single state, it is necessary that each state be capable of merging with all other states. Requirement (2) also states that no more than one state may contain a stable state in any given input column of the flow matrix. If two states contain stable states in the same column, then they must be considered as redundant states. From this, it is seen that a flow matrix with "n" different inputs may have no more than "n" different states merged as a single state.

To facilitate optimum merging of states, Huffman (3), (5) introduces a merger diagram. The flow matrix of Figure 5(a) has been examined for mergers, and the results have been indicated on the merger diagram of Figure 5(b). The numbers of the merger diagram represent the six different states of the flow matrix, and a line connecting two state numbers indicates that these two states may be merged into a single state.

X_1	X_2	X_3	X_4	Z_1
1	3	-	6	01
-	3	2	6	10
1	3	5	-	01
4	3	-	6	10
-	3	5	6	01
4	-	2	6	10

Figure 5(a). A Flow Matrix to Be Reduced Through Merging

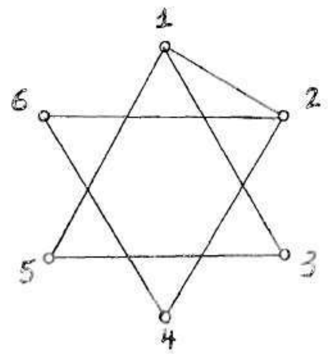

Figure 5(b). Merger Diagram for Figure 5(a)

The optimum merger is obtained after examining the merger diagram. If states 1, 3, and 5 are merged as a single state, and states 2, 4, and 6 are merged as a single state, the minimum flow matrix is then found. The two state flow matrix is shown in Figure 5(c). The excitation matrix is formed by the required states in the same manner used in combining redundant states. The output matrix shows the outputs associated with each stable state of the excitation matrix.

X_1	X_2	X_3	X_4
1	3	5	6
4	3	2	6

Excitation Matrix

X_1	X_2	X_3	X_4
01	01	01	-
10	-	10	10

Output Matrix

Figure 5(c). Excitation and Output Matrix for the Example of Figure 5(a)

Optimum Reductions

As mentioned previously, dependency often exists among the redundant states. Equally important, however, is the correlation necessary between reduction of a flow matrix by removing redundant states and reduction by merging states.

An example given by Caldwell (5) which illustrates the dependency among the two reduction procedures is shown in Figure 6(a). In this example, it is possible to combine redundant states in two different ways. The redundant states are pairs 1 and 6, 2 and 8, and 2 and 6. The dependency among these redundant states is different from the dependency illustrated by Figure 3(a). There, one redundant pair depended upon the other pair being redundant before the pair of states could be combined. In Figure 6(a), all three pairs are redundant when examined separately; however, if pair 2 and 6 are combined as redundant states, then the other two pairs cease to be redundant.

X_1	X_2	X_3	X_4	Z_1
1	-	4	7	10
2	3	5	-	10
8	3	5	-	11
-	9	4	7	01
-	3	5	7	01
6	3	-	7	10
1	-	4	7	00
8	-	5	10	10
6	9	4	-	00
2	-	5	10	11

Figure 6(a). A Flow Matrix Illustrating Dependency Between Redundant States and Mergers

X_1	X_2	X_3	X_4	Z_1
1	-	4	7	10
2	3	5	7	10
8	3	5	-	11
-	9	4	7	01
-	3	5	7	01
1	-	4	7	00
8	-	5	10	10
2	9	4	-	00
2	-	5	10	11

Figure 6(b). Flow Matrix Resulting When Redundant States Two and Six of Figure 6(a) Are Combined

Another possibility is to combine pair 1 and 6, and then pair 2 and 8. Upon doing this, Figure 6(c) shows no other redundant states. In this example, it was necessary to select which redundant states are to be removed since eliminating all three redundant pairs is impossible.

X_1	X_2	X_3	X_4	Z_1
1	3	4	7	10
2	3	5	10	10
2	3	5	-	11
-	9	4	7	01
-	3	5	7	01
1	-	4	7	00
1	9	4	-	01
2	-	5	10	11

Figure 6(c). Flow Matrix Resulting When Redundant States 1 and 6, and 2 and 8 of Figure 6(a) Are Combined

The dependency between mergers and the combination of redundant states is illustrated in Figures 6(d) and 6(e); i.e., the merger diagrams for Figures 6(b) and 6(c) respectively. The flow matrix of Figure 6(c) may be merged into a four state matrix, while Figure 6(b) will only reduce to a five state matrix.

The optimum combination of redundant states will not always yield the minimum flow matrix as was the case of the previous example. As flow matrices increase in size and complexity, it becomes increasingly difficult

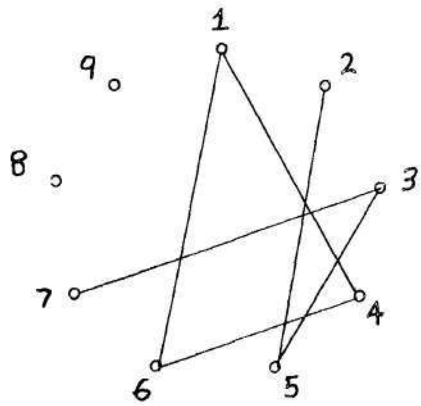

Figure 6(d). Merger Diagram for Figure 6(b)

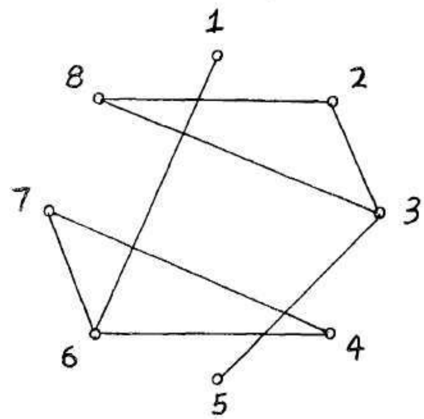

Figure 6(e). Merger Diagram for Figure 6(c)

to determine what redundant states exist and how they may be combined to yield the minimum flow matrix after merging.

CHAPTER III

GINSBURG'S REDUCTION PROCEDURE

Introduction

A second method for the reduction of sequential switching circuits to be considered was developed by S. Ginsburg (11). This method will be presented here in outline form, and the portions of the procedure which pertain to the discussion in the following chapters will be explained in detail.

It is important to note that Ginsburg does not merge a flow matrix in the sense described by Huffman, but actually rewrites the flow matrix in reduced form.

Ginsburg's Flow Matrix Format

Ginsburg's flow matrix format is illustrated in Figure 7. In this figure, the possible inputs are X_1, X_2, and X_3, and the states of the circuit are listed as numbers 1 through 6. Each input and state combination will describe the next circuit action with two numbers. These numbers are (1) the next state of the circuit (shown without parentheses) and (2) the next output of the circuit (shown enclosed in parentheses). According to this notation, if the example of Figure 7 should be in state 1, and input X_3 occurs, the circuit will advance to state 6 while the output changes to output 3.

Either or both of the next-state, next-output designations may be omitted from the flow matrix. An examination of state 3 of Figure 7

States	X_1	X_2	X_3
1	(1)	–	6(3)
2	–	5(2)	–
3	(1)	–	4
4	–	(1)	3
5	2(2)	–	(1)
6	–	1(1)	(2)

Figure 7. A Flow Matrix Illustrating Ginsburg's Notation

shows three possibilities. If input X_1 occurs, the circuit will remain in state 3 while output 1 occurs; if input X_2 occurs, the circuit will remain in state 3 without any requirements upon the output; and if input X_3 occurs, the circuit will advance to state 4 without any requirements upon the output.

Ginsburg allows another degree of freedom in the flow matrix in that the output is not specified for each state of the circuit. The same consideration may also be given to Huffman's flow matrix as explained in Appendix II.

Outline of Synthesis Procedure

An outline of Ginsburg's synthesis procedure is as follows:

(1) A particular lower bound on the minimum number of states attainable in the reduced flow matrix is determined.

(2) Every possible circuit with this minimum number of states is tested to see if it satisfies the original circuit requirements.

(3) If all possibilities of step (2) are exhausted without finding a reduced flow matrix the minimum number is increased by one, and step (2) is repeated.

(4) The process is terminated when a reduced matrix is found which satisfies the requirements of the original circuits.

In his paper, Ginsburg explains the above steps with the aid of several examples. It is also noted that while step (1) is automatic and is easily programmed for computer reduction application, the execution of steps (2) and (3) depends upon the enumeration and selection of numerous alternative paths and may not be programmed in its present form.

The following chapters will be concerned with the execution of step (1) which will now be explained in a less formal language than that used by Ginsburg. Ginsburg's examples are discussed as examples seven, eight, and nine of Appendix III.

Compatible and Incompatible States

For any flow matrix, any arbitrary choice of two states (either different states or the same state) may be classified as being compatible or incompatible. A compatible state-pair is a twosome of two states which may possibly be combined or merged in a reduced flow matrix. Incompatible state-pairs are then all pairs of remaining states after the compatible state-pairs have been determined.

The compatible state-pairs are determined through a repeated process of elimination of possible compatible state-pairs. Here, a possible compatible state-pair is a pair of states which are compatible when the pair is examined only with respect to the next output designations. That is, two states form a possible compatible state-pair if for every input, the new outputs of the two states are either the same or if one, or both, of the outputs are unspecified.

From the definition of a possible compatible state-pair, it is obvious that any state is compatible with itself. The subset of all pairs of the same state number is given the notation H, where

$$H \equiv \bigcup_{i=1}^{n} (i,i) \text{ .}$$

Here i is the state number and n is the total number of states in the flow matrix. The complete set of all possible compatible state-pairs is the union of H with the subset of all compatible state-pairs of unlike state numbers.

If the flow matrix of Figure 8 is examined, the set of all possible compatible state-pairs is given by

$$P_1 \equiv H \bigcup \{(1,2),\ (1,4),\ (1,5),\ (2,3),\ (2,5),\ (2,6),\ (6,8), \\ (2,7),\ (2,8),\ (3,4),\ (3,6),\ (3,8),\ (4,5),\ (4,7)\} \text{ .} \qquad (3)$$

Here, P_1^* denotes the set of possible compatible state-pairs determined in the first examination.

It is now necessary to eliminate any of the above possible compatible state-pairs which may prove to be incompatible upon further

---

*State-pairs are not to be confused with orders pairs. However, for convenience in the analysis of the flow matrices, the state-pairs will always be listed with the lowest state number in the first position.

States	X_1	X_2	X_3
1	2(1)	–	3(2)
2	–	4(2)	–
3	7(2)	–	6(1)
4	–	2(1)	–
5	5(1)	–	3(2)
6	–	1(2)	7(1)
7	6(2)	–	1(2)
8	–	–	2(1)

Figure 8. A Flow Matrix to Be Examined for Compatible States

examination. The rule to follow is to eliminate any state-pair which, for any input, advances to a state-pair which is not listed in the set of possible compatible states. For example, consider pair (3,6) when input X_3 occurs. From Figure 8, it is seen that state 3 advances to state 6 and state 6 will advance to state 7. However, the pair (6,7) is not listed in set P_1; therefore, pair (3,6) is incompatible. With the elimination of pair (3,6) the possible compatible state-pairs are now given by

$$P_2 \equiv H \bigcup \{(1,2),\ (1,4),\ (1,5),\ (2,3),\ (2,5),\ (2,6),\ (2,7),\ (2,8),\ (3,4),\ (3,8),\ (4,5),\ (4,7),\ (6,8)\}\ . \tag{4}$$

P_2 denotes the set of possible compatible state-pairs determined in the second examination.

P_2 is examined in the same manner as P_1 was examined, and any other incompatible pairs are eliminated. All state-pairs of P_2 which pass the examination are rewritten as P_3. This process is continued until an examination shows no pairs which can be eliminated. In this, example,

$$P_3 \equiv P_2 ; \tag{5}$$

therefore, the process terminates and all pairs listed under P_2 are the compatible state-pairs. The set of incompatible state-pairs are all pairs of states not listed under P_2. These are

$$\{(1,3),\ (1,6),\ (1,7),\ (1,8),\ (2,4),\ (3,5),\ (3,6),\ (3,7),$$
$$(4,6),\ (4,8),\ (5,6),\ (5,7),\ (5,8),\ (6,7),\ (7,8)\} . \tag{6}$$

<u>The Theoretical Minimum Attainable Flow Matrix</u>

To determine the theoretical minimum number of states attainable in a reduced flow matrix, it is necessary to examine the set of incompatible state-pairs of the flow matrix.

The set of incompatible state-pairs is first arranged into subsets according to the lowest number of each pair; that is, the subset of all pairs of (6) which contain a 1 would be

$$\{(1,3),\ (1,6),\ (1,7),\ (1,8)\} . \tag{7}$$

A set is then formed using the highest number of each pair of the subsets. For (7) this set is

$$\{1,3,6,7,8\} . \tag{8}$$

Since for the flow matrix under consideration, there are seven different numbers which are the lowest number of a pair of incompatible states as given by expression (6), there will be seven sets of states. These seven sets are shown in Figure 9.

Set No.	States	Number of States
1	{1,3,6,7,8}	5
2	{2,4}	2
3	{3,5,6,7}	4
4	{4,6,8}	3
5	{5,6,7,8}	4
6	{6,7}	2
7	{7,8}	2

Figure 9. Ordering of the Incompatible States of Figure 8 to Determine the Number of States in the Minimum Attainable Flow Matrix

The number of states in the largest set (in this case set 1 of Figure 9), theoretically represents the minimum number of states in the reduced flow matrix. However, although each member of a set in Figure 9 is incompatible with a common member of the set (in this case the member of the set which is the same as the set number), not all members are incompatible with every other member in the set. An examination is made of each set beginning with the lowest state number of the set. If any state is found to be listed as a compatible state-pair with any state of the set, then one of these states is eliminated from the set (to be consistent, the larger of the state numbers is eliminated in each case).* The compatible state-pairs for Figure 9 are given by Equality (4).

An examination of set 1 of Figure 9 reveals that states 1, 3, 6, or 7 will not form any compatible state-pairs; however, state 8 is compatible with state 3. Therefore, state 8 is removed from the first set of Figure 9.

A further examination of set 1 reveals that no other states may be removed, and the total number of states in set 1 is four. Since no other set contains more than four states, the examination is concluded.

It has now been determined that the reduced flow matrix of Figure 8 will contain a minimum of four states. The theoretical minimum is a lower bound for a flow matrix which is to be reduced and is not necessarily attainable in each problem.

---

*Readers are referred to Ginsburg's article (11) for his explanation and examples.

CHAPTER IV

BASIS FOR A NEW REDUCTION PROCEDURE

Introduction

At this point, a new procedure for reducing a flow matrix to minimum form is introduced. The method to be presented is based in part upon the contributions of Huffman and Ginsburg as described in the previous chapters; however, the applications of the contributions differ from that described by either author.

The procedure is presented in a manner which will allow it to be programmed for a computer solution, that is, concise rules will be given for each step of the procedure. Except for differences in the initial examination and the final construction of the reduced flow matrix, the procedure will be the same regardless of which type of flow matrix is to be reduced.

Compatible and Incompatible States for Huffman's Flow Matrix

It is necessary to extend the concepts of Ginsburg so as to be able to determine the compatible state-pairs for Huffman's flow matrix and from this determine the theoretical minimum attainable reduced flow matrix.

Figure 10 shows a flow matrix which will be examined for possible compatible state-pairs. Two states of Huffman's flow matrix will form a possible compatible state-pair if either

(1) their outputs are the same, or

(2) their outputs are different, but both states do not simultaneously have stable states occurring within the same input column.

According to these rules, the possible compatible state-pairs for Figure 10 are:

$$P_1 \equiv H \bigcup \{(1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (1,8), (2,3), (2,4), (2,5), (2,6), (2,7), (3,5), (3,6), (3,7), (3,8), (4,5), (4,6), (4,7), (4,8), (5,6), (5,7), (5,8), (6,7), (6,8)\} . \quad (9)$$

Again, to obtain the compatible state-pairs, it is necessary to eliminate any of the possible compatible state-pairs which may prove to be incompatible upon further examination.

States	X_1	X_2	X_3	X_4	Z_1
1	1	3	-	7	0
2	2	3	-	7	0
3	1	3	6	-	1
4	2	4	5	-	0
5	-	4	5	8	0
6	-	4	6	8	0
7	2	-	5	7	1
8	1	-	6	8	0

Figure 10. A Flow Matrix of Huffman's Notation to Be Examined for Compatible State-Pairs

A state-pair is incompatible if for any input either of the following is true:

(1) One state contains a stable state number while the second state contains an unstable state number so that the first state and the state containing the stable state number, corresponding to the unstable state number, are not listed as a possible compatible state-pair.

(2) Both states contain unstable state numbers such that the states which contain the corresponding stable state numbers are not listed as a possible compatible state-pair.

As an example of a violation of rule (1), consider pair (1,4) from Equality (9). For input X_2, state 4 contains stable state number 4. State 1 contains unstable state number 3 which requires that state 1 advance to the state containing stable state 3 when input X_2 occurs (namely state 3). However, (3,4) is not listed in Equality (9); hence (1,4) is actually incompatible. Possible compatible state-pair (1,5) will be rejected under rule (2) when examined with respect to input X_2.

The process of elimination is repeated until a complete examination results in no further eliminations. The pairs remaining are then the compatible state-pairs. For Figure 10 the elimination is as follows:

$$P_2 \equiv H \cup \{(1,2),\ (1,3),\ (1,7),\ (2,3),\ (2,7),\ (3,7),(4,5),\ (4,6),\ (4,7),\ (4,8),\ (5,6),(5,8),\ (6,8)\}\ . \tag{10}$$

$$P_3 \equiv P_2\ . \tag{11}$$

The examination terminates with P_3, and the compatible state-pairs for Figure 10 are those given by P_2 of Equality (10).

The Theoretical Minimum Attainable Flow Matrix

Determination of the theoretical minimum number of states attainable in a reduced flow matrix of Huffman's type follows the same procedure as for Ginsburg's flow matrix. For the example of Figure 10, the incompatible state-pairs are:

$$\{(1,4),\ (1,5),\ (1,6),\ (1,8),\ (2,4),\ (2,5),\ (2,6),\ (2,8),\ (3,4),\ (3,5),\ (3,6),\ (3,8),\ (5,7),\ (6,7)\}\ . \tag{12}$$

The incompatible state-pairs are grouped into sets according to the lowest state number of each pair in Figure 11(a). All sets of Figure 11(a) are then examined and any element of a set which is found to be compatible with an element of that set with a lower number [as given by P_2 of Equality (10)] is removed from the set. Figure 11(b) shows the sets after

Set Number	Elements
1	$\{1,4,5,6,8\}$
2	$\{2,4,5,6,8\}$
3	$\{3,4,5,6,8\}$
4	$\{5,6\}$
5	$\{6,7\}$

Figure 11(a). Sets of Possible Incompatible States Numbers to Be Examined in Order to Remove any Compatible State Numbers

Set Number	Elements
1	$\{1,4\}$
2	$\{2,4\}$
3	$\{3,4\}$
4	$\{5,7\}$
5	$\{6,7\}$

Figure 11(b). Sets of Incompatible State Numbers After Removal of All Compatible State Numbers they have been examined. From Figure 11(b) it is seen that the largest set contains two elements; therefore, the theoretical minimum attainable reduced flow matrix for the example of Figure 10 will contain two states.

Merger-Restriction Table for Ginsburg's Flow Matrix

Assuming that the method of obtaining the compatible state-pairs for Ginsburg's flow matrix has been applied, it is now possible to examine the flow matrix and determine any restrictions which may exist upon the pairs of compatible state-pairs.

For Ginsburg's Flow Matrix of Figure 8, the compatible state-pairs were determined to be

$$P_2 \equiv H \cup \{(1,2),\ (1,4),\ (1,5),\ (2,3),\ (2,5),\ (2,6),\ (2,7),\ (2,8),\ (3,4),\ (3,8),\ (4,5),\ (4,7),\ (6,8)\} \quad (4)$$

Application of the process of eliminating possible compatible states which are incompatible to P_2 results in no eliminations; however, it is noted that certain pairs are actaully restrictions upon other pairs. For example, if pair (1,5) is examined with respect to input X_1 in Figure 8, it is seen that state 1 advances to state 2 while state 5 remains in state 5. Since pair (2,5) is listed as being compatible, pair (1,5) is therefore compatible; however, (2,5) is a restriction upon pair (1,5).

If every compatible pair listed under P_2 is examined in this same way, then any restrictions associated directly with a compatible state-pair may be recorded. Figure 12 shows such a table for the matrix

Possible Mergers	Restrictions
(1,2)	-
(1,4)	-
(1,5)	(2,5)
(2,3)	-
(2,5)	-
(2,6)	(1,4)
(2,7)	-
(2,8)	-
(3,4)	-
(3,8)	(2,6)
(4,5)	-
(4,7)	-
(6,8)	(1,4), (2,7)

Figure 12. Merger-Restriction Table for Ginsburg's Flow Matrix of Figure 8

of Figure 8. The first column lists all of the compatible state-pairs under the title "Possible Mergers," while the second column lists any restrictions associated with a state-pair under the title "Restrictions."* A listing such as shown in Figure 12 will be known as a merger-restriction table.

---

*Hereafter, a compatible state-pair shall also be identified as a possible merger. It is noted that state-pairs listed as restrictions are also listed as possible mergers. The distinction between considering a state-pair as a possible merger and considering it as a restriction will be explained later.

Merger Restriction Table for Huffman's Flow Matrix

A merger restriction table may also be found for Huffman's flow matrix. The determination of the compatible state-pairs has been explained, and the restriction will be determined by applying the elimination rules of possible compatible state-pairs to the compatible state-pairs.

For the example of Figure 10, the compatible state-pairs were found to be

$$P_2 \equiv H \cup \{(1,2),\ (1,3),\ (1,7),\ (2,3),\ (2,7),\ (3,7),\ (4,5),\ (4,6),\ (4,7),\ (4,8),\ (5,6),\ (5,8),\ (6,8)\}\ . \tag{10}$$

As an example of the determination of restrictions, consider state-pair (1,7) as listed above. If input column X_1 in Figure 10 is examined, it is seen that state 1 remains unchanged while state 7 advances to state 2. State-pair (1,2) is a compatible state-pair, and is also a restriction upon pair (1,7).

Examination of all compatible state-pairs for Figure 10 will yield the merger-restriction table of Figure 13.

Significance of the Merger-Restriction Table

There is an equivalence between the state-pairs of the merger-restriction table and the mergers and redundant states of Huffman's flow matrix (3),(5). The equivalence is that any state-pair from the merger-restriction table which is listed as a restriction would also be

Possible Mergers	Restrictions
(1,2)	–
(1,3)	–
(1,7)	(1,2)
(2,3)	(1,2)
(2,7)	–
(3,7)	(1,2), (5,6)
(4,5)	–
(4,6)	(5,6)
(4,7)	–
(4,8)	(1,2), (5,6)
(5,6)	–
(5,8)	(5,6)
(6,8)	–

Figure 13. Merger-Restriction Table for Huffman's Flow Matrix of Figure 10

considered by Huffman to be a redundant state. Redundant states, however, are now always considered as restrictions as explained in example three of Appendix III. The possible merger column of the table lists all the state-pairs which could possibly be merged or combined in the reduced flow matrix.

The merger-restriction table completely characterizes the flow matrix it represents. That is, all possible reduced flow matrices may be determined from the merger-restriction table. It should be noted that, given a merger-restriction table, it is impossible to determine whether the table was formed from Huffman's or Ginsburg's flow matrix. This is consistent with the fact that a given sequential circuit action could be described with either or both flow matrices. In general, it is suspected that merger-restriction tables exist for any type of flow matrix which describes a sequential circuit.

Rules of Merging and Combining

Any state-pair listed in the merger-restriction table could theoretically be merged or combined in a reduced version of the original flow matrix. More generally, however, it is necessary to classify each state-pair according to one of three sets or groups when considering the overall relationship between all state-pairs. These divisions are:

(1) The set of combinations.

(2) The set of mergers.

(3) The set of rejections.

Each of these classifications may be explained by considering the state-pairs which they include.

Combinations

If a state-pair is considered as a restriction in the merger-restriction table, then this requires that the pair of states be combined in the reduced flow matrix; hence, the set of combinations will be a subset of the set of restrictions. The word subset is used here since some state-pairs may not be considered as a combination although they are listed as a restriction in the merger-restriction table.

If the example of Figure 6(a) is considered, the restrictions for the merger-restriction table would be state-pairs (1,6), (2,6) and (2,8). However, the combinations which yielded the minimum flow matrix were (1,6) and (2,8) in Figure 6(c).

In this example, it was impossible to combine all restrictions in the final flow matrix; however, in other problems it may be necessary to ignore a restriction, which is possible, in order to achieve the minimum flow matrix.

Mergers

A state-pair may be considered as a merger if it is listed as a possible merger in the merger-restriction table. A merger here means a pair of states which could possibly (but not necessarily) be merged in a reduced flow matrix.

As an illustration, consider the flow matrix of Figure 5(a). There the state-pairs considered as mergers were

$$\{(1,2),\ (1,3),\ (1,5),\ (2,4),\ (2,6),\ (3,5),\ (4,6)\}\ . \tag{13}$$

In the final reduced flow matrix, however, state-pair (1,2) was not merged in order to achieve the minimum flow matrix. The set of mergers will be a subset of all possible mergers and may even include state-pairs which are also listed as restrictions, but which have not be considered as combinations.

Rejections

Finally, the set of rejections will include all state-pairs which cannot be included in either of the two previous classifications. Thus, for the example of Figure 6(a), 6(c), state-pair (2,6) is a rejection since it was not used as a combination or as a merger.

In addition, any possible merger which has associated with it one or more restrictions must be considered as a rejection unless all of its restrictions have been classified as combinations.

Rules for Determining Classification of State-Pairs

When considering an individual state-pair as to its classification in a reduced flow matrix, it is necessary to also consider what classifications have previously been given to other state-pairs. The following rules must be applied in determining this classification. Given any pair of states A and B, then,

(1) If A and B are to be considered as a combination, or as a merger, then they must be listed as a state-pair in the possible merger column of the merger-restriction table, and any restrictions associated with this pair must be combined in the reduced flow matrix.

(2) If state A is to be capable of merging with state B, then state A must also be capable of merging with all states which have previously been indicated as being combined with state B.

(3) If state A is to be combined with state B, then state A must be capable of

- (a) merging with all states which have previously been indicated as capable of merging with state B;
- (b) combining with all states which have previously been indicated as combining with state B.

Requirements (2) and (3) above must also be satisfied by state B with respect to state A.

Construction of a Reduced Flow Matrix

The rules set forth in the preceding section and the merger-restriction table of Figure 13 will now be used to construct a reduced flow matrix for Figure 10.

The procedure to be employed is to "arbitrarily" select a possible merger for examination. If all requirements upon this possible merger can be fulfilled, then the possible merger is accepted; otherwise, it will be rejected. When all possible mergers have been tested, then the optimum mergers will be selected so as to yield the minimum flow matrix.

Figure 14(a) shows the basic merger diagram as described by Huffman (3), where the numbers represent the state numbers of the flow matrix of Figure 10. A step-by-step construction of the merger diagram would be as follows:

(1) The possible merger under consideration is (1,7) and the restrictions upon this pair is (1,2). Pair (1,7) is indicated as capable of being merged in Figure 14(b). It is now necessary to examine restriction (1,2). From rule 3(b), state 2 may be combined with state 1 only if state 2 will merge with state 7. From Figure 13, (2,7) is listed as a possible merger; therefore, Figure 14(c) shows the diagram with the required combination of (1,2). From the merger-restriction table, it is seen that neither (1,7) nor (1,2) have any restrictions. If any restrictions had existed, it would now be necessary to examine each restriction in turn and indicate its acceptance on the merger diagram [as was (1,2)] if it satisfies all the rules. Also, any new restrictions which are found must be tested. If any restrictions should fail a test,

then this prevents the acceptance of merger (1,7), and it, along with any combinations performed on its behalf, must be removed from the merger diagram.

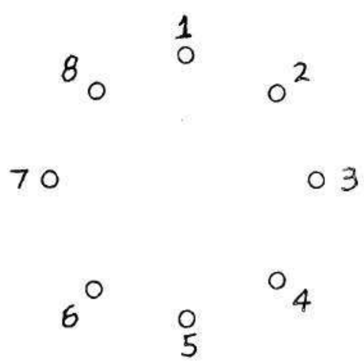

Figure 14(a). Basic Merger Diagram

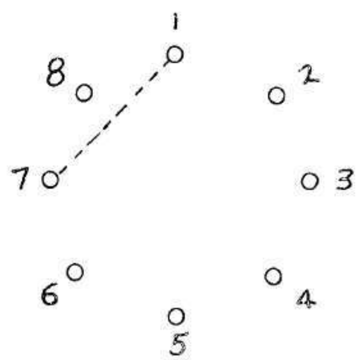

Figure 14(b). Modified Merger Diagram for Examination of Possible Merger (1,7)

(2) The possible merger under consideration is (2,3) and the restriction upon this pair is (1,2). Since (1,2) has previously been combined, it is only necessary to test (2,3) with rule (2). For state 3, to merge with state 1, it is necessary that (1,3) be capable of merging. In Figure 13, pair (1,3) is listed as a possible merger with no restrictions; thus, (1,3) and (2,3) may both be accepted as possible mergers and are so indicated in Figure 14(d).

(3) The possible merger under consideration is (3,7), and the restrictions upon this pair are (1,2) and (5,6). The mergers and restrictions will be checked against the merger diagram of Figure 14(e).*

*While a possible merger is being tested, it is indicated as a dashed line on the merger diagram. At the same time, the states which are being tested as a combination are indicated with a check. If the possible merger under consideration is accepted, then the state numbers of

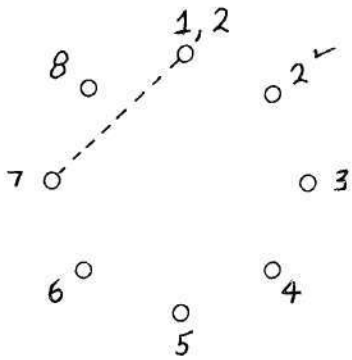

Figure 14(c). Modified Merger Diagram for Examination of Possible Merger (1,7) and Combination (1,2)

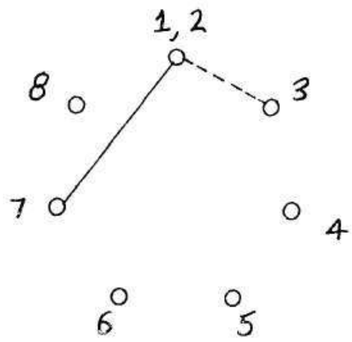

Figure 14(d). Modified Merger Diagram for Examination of Possible Merger (1,3)

No states have previously been combined with states 3 and 7; therefore, these numbers are shown connected with a dashed line in Figure 14(f), and no new restrictions are found. Restriction (1,2) has been combined previously, and it is now only necessary to check restriction (5,6) with rule (3). Combination (5,6) is possible, and no new restrictions are found. Figure 14(f) is thus acceptable.

(4) The possible merger under consideration is (4,6) and the restriction upon this pair is (5,6). Since (5,6) has been combined previously, it is necessary only that (4,6) satisfy rule (2); that is, from Figure 14(g), state 4 must be capable of also merging with state 5. The pair (4,5) is listed in Figure 13 with no restirctions; thus all restrictions have been satisfied, and all rules for possible merger (4,6) have been passed. Figure 14(g) is accepted.

---

the possible merger are connected with a solid line, and the state numbers which have been combined with other state numbers are removed from the diagram. If a trial should fail, then the merger diagram is returned to its condition before the trial.

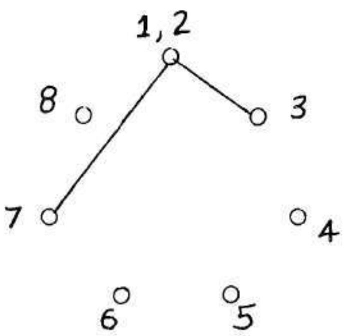

Figure 14(e). Modified Merger Diagram for Examination of Possible Merger (3,7)

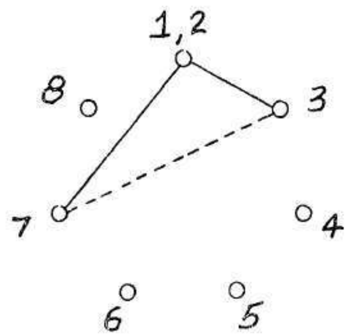

Figure 14 (f). Modified Merger Diagram for Examination of Possible Merger (3,7) and Combination (5,6)

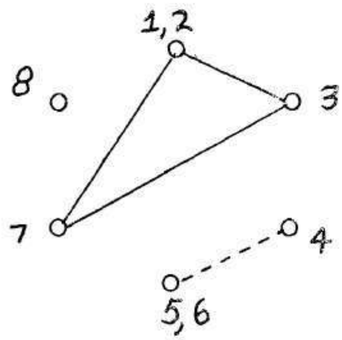

Figure 14(g). Modified Merger Diagram for Examination of Possible Merger (4,6)

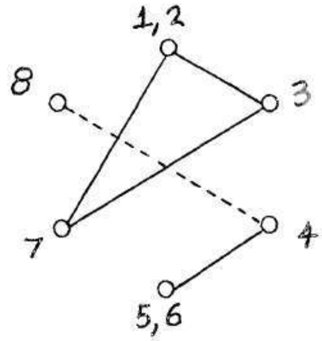

Figure 14(h). Modified Merger Diagram for Examination of Possible Merger (4,8)

(5) The possible merger under consideration is (4,8) and the restrictions are (1,2), and (5,6). All restrictions have been combined previously and (4,8) satisfy rule (1), with no new restrictions being found; therefore, (4,8) is accepted in Figure 14(h).

(6) The possible merger under consideration is (5,8) and the restriction is (5,6). The restriction has been combined previously;

thus it is only necessary to examine the merger of (5,8) by rule (1). From Figure 14(i), it is seen that possible merger (6,8) is then required. Merger (6,8) is possible from Figure 13, and no new restrictions are found. Therefore, Figure 14(i) is accepted.

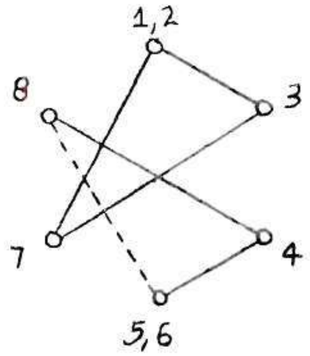

Figure 14(i). Modified Merger Diagram for Examination of Possible Merger (5,8)

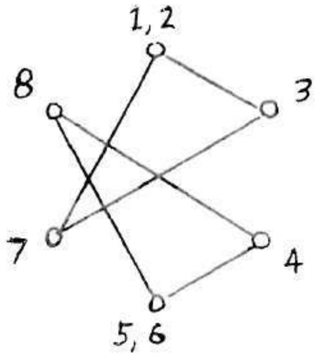

Figure 14(j). Modified Merger Diagram for Flow Matrix of Figure 13

At this point, all possible mergers have been attempted for the example of Figure 13, and all trials were successful. Each trial consisted of testing a possible merger and all associated restrictions as mergers and combinations, respectively. The order of testing each possible merger and its associated restrictions in each trial is unimportant since it is necessary that each state-pair pass all required tests, both with respect to all mergers and combinations previously accepted and with respect to the merger and combinations of that trial. Again, if the possible merger or any restriction of a trial should fail to satisfy the necessary rules, then neither the merger nor any of the restrictions can be accepted.

The final merger diagram is shown in Figure 14(j), where the optimum combination of states is selected as previously described by Huffman (3), and Caldwell (5). In this example, the reduced flow matrix will consist of two states; that is, state 1 will be made of states 1, 2, 3, and 7 of the original flow matrix (Figure 10), while state 2 will consist of states 4, 5, 6, and 8 of the original flow matrix. Figures 15(a) and 15 (b) show the excitation and output matrices, respectively, as determined above.

State	X_1	X_2	X_3	X_4
1	1	3	4	7
2	1	4	5	7

Figure 15(a). Excitation Matrix for the Reduced Flow Matrix of the Example of Figure 10

State	X_1	X_2	X_3	X_4
1	0	1	-	1
2	-	0	0	-

Figure 15(b). Output for the Reduced Flow Matrix of the Example of Figure 10

From the study of incompatible states for the flow matrix of Figure 10, it was determined that the theoretical minimum flow matrix would contain two states. A two-state flow matrix has been found for Figure 10; therefore, the minimum reduced flow matrix has been determined.

CHAPTER V

THE REDUCTION PROCEDURE

Dependency Among the Restrictions

It is possible for either of two conditions to exist upon a restriction listed in a merger-restriction table. That is, when the restriction is listed in the possible merger column, it may be listed as a possible merger with no restrictions upon itself, or it may be listed as a possible merger with one or more restrictions associated with itself. These two conditions partially express the independency or dependency, respectively, of the restrictions.

The examples of Figures 3(a) and 6(a) illustrate two types of dependencies which may exist between the redundant states of Huffman's flow matrices. For Figure 3(a) it was found that stable states 3 and 10 were redundant only if stable states 4 and 11 were redundant. However, 4 and 11 were redundant only if 3 and 10 were redundant; therefore, it was possible to conclude that both pairs were simultaneously redundant. It would have been incorrect to treat one pair as being redundant and ignore the pair; hence, there exists a dependency between the two pairs of redundant states.

For Figure 6(a), it was found that the acceptance of one pair of redundant states prevented the acceptance of another redundant state; hence, a choice as to which pair to accept was necessary. However, the proper choice had to be made in order to achieve the minimum reduced flow matrix.

The reduction example, given in the previous chapter, arbitrarily selected possible mergers from the merger-restriction table, and accepted all possible mergers for which the restrictions could be satisfied. While this random selection method would be satisfactory for the type of dependency between restrictions, as illustrated by the example of Figure 3(a), it is possible that the restrictions satisfied for the example of Figure 6(a) would not be the restrictions which would yield the minimum flow matrix.*

Restriction-Dependency Table

A restriction-dependency table is now constructed as an aid in enumerating the possible reduced flow matrices and to simplify the determination of the dependencies which may exist among the restrictions. Figure 16 shows the restriction-dependency table for the merger-restriction table of Figure 12. All of the restrictions of the table of Figure 12 are listed in the restriction column of Figure 16. Next, each restriction of Figure 12 is examined when it is listed as a possible merger. If the state-pair has any restrictions associated with itself when it is listed as a possible merger, then the state-pairs which form the restrictions are listed in the dependency column of the restriction dependency column.

From a restriction-dependency table, it is easily determined which restrictions may be combined in a reduced flow matrix.

---

*The minimum reduced flow matrix could be found in all cases if every permutation (of selecting possible mergers for trial) were considered. The minimum flow matrix could then be chosen from all the reduced flow matrices found.

Position	Restriction	Dependency
a	(1,4)	-
b	(2,5)	-
c	(2,6)	(1,4)
d	(2,7)	-

Figure 16. Restriction-Dependency Table for the Merger-Restriction Table of Figure 12

Outline of Reduction Procedure

A reduction procedure based upon the material discussed so far will now be outlined. The procedure will result in the determination of the minimum attainable flow matrix through merging, and is applicable to both types of flow matrices discussed so far. In outline form, the procedure is as follows:

(1) Construct the merger-restriction table for the flow matrix to be reduced.

(2) Determine the number of states in the theoretical minimum flow matrix.

(3) Construct the restriction-dependency table.

(4) Combine one possible subset of the set of restrictions.

(5) Determine the number of states existing in a flow matrix based upon the subset of restrictions under consideration.

(6) Compare the number of states as obtained in step (5) against the theoretical minimum number determined in step (2).

(7) If the theoretical minimum number of states has been found, the procedure terminates with the construction of the reduced flow matrix; otherwise repeat steps (4), (5), and (6) with a different subset of restrictions.

The procedure will continue until one subset under examination yields the theoretical minimum number of states in the reduced flow matrix, or until all subsets of the set of restrictions have been examined. In the latter situation, the theoretical minimum matrix is unattainable, and the minimum matrix which was found is accepted.

In step (6) it is possible to compare the number of states in each reduced matrix with the theoretical minimum number of states based upon a percentage reduction rather than minimum reduction. For example, if the theoretical minimum attainable matrix has T states, the original matrix has M states, P is the percentage reduction required, and R_1 is the largest integer less than R where

$$R = \left(1 - \frac{P}{100}\right)\left(M + T\right) \tag{14}$$

The first reduced flow matrix having R_I (or less) states is then accepted.

Step (5) uses rules (1) and (2) of the previous chapter, and the merger-restriction table to construct a merger diagram. The states which have been combined, as indicated in step (4), are indicated on the merger diagram. Each state-pair listed as a possible merger in the merger-restriction table is then examined. If a state-pair either has no restriction or if the restrictions upon a state-pair has been combined, the state-pair is then tested against the merger diagram to determine whether it may be indicated as a possible merger. After all possible mergers have been examined, the minimum number of states in a reduced flow matrix is then determined and compared against the theoretical number of states in step (6).

Determination of the Subsets of the Set of Restrictions

There are 2^n possible subsets of a set containing n elements; hence, for the four restrictions of Figure 17, there are $2^4 = 16$ theoretically possible reduced flow matrices. However, because of the dependencies which may exist among the restrictions, not all subsets are possible. A binary counting technique is employed in order that the selection of the subsets be systematic and complete. Rules 1 and 3(b) will be employed to test each element of a subset.

Each restriction of the restriction-dependency table is assigned a position in a binary number. A binary count is then made from zero to 2^n where n is four in this example. For each binary number, if a position contains a one, then the restriction corresponding to that position is an element of the subset corresponding to that binary number. Figure 17 shows the binary count for the restrictions of Figure 16.

Before any subset if accepted, it is necessary that each element of the subset be combined according to rule 3(b), and that any dependencies existing upon the elements also be combined. If this is impossible, the subset if rejected and a new subset is chosen. As an example, consider the subset denoted by 0111. This subset consists of the elements (2,6), (2,5), and (1,4). The only dependency is state-pair (1,4) which

Restrictions (2,7)	(2,6)	(2,5)	(1,4)	Subsets Possible
0	0	0	0	✓
0	0	0	1	✓
0	0	1	0	✓
0	0	1	1	✓
0	1	0	0	✓
0	1	0	1	✓
0	1	1	0	-
0	1	1	1	-
1	0	0	0	✓
1	0	0	1	✓
1	0	1	0	-
1	0	1	1	-
1	1	0	0	-
1	1	0	1	-
1	1	1	0	-
1	1	1	1	-

Figure 17. Binary Count in Order to Determine the Subsets of the Set of Restrictions for Example in Figure 16

is also listed as an element of the subset; therefore, the subset is complete. Figure 18 shows the merger diagram when (2,6) and (1,4) have been combined [since they both satisfy rule 3(b)]. However, when (2,5) is combined, it is necessary that (5,6) also be allowed according to rule 3(b). Examination of Figure 12 shows that (5,6) violates rule 1;

rule 3(b). Examination of Figure 12 shows that (5,6) violates rule 1; hence it is impossible. Thus this subset is impossible, and must be rejected.

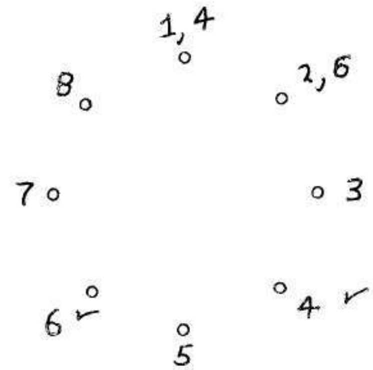

Figure 18. Examination of Subset $\{(2,6),\ (2,5),\ (1,4)\}$ from Figure 17

If all subsets of Figure 16 are examined, it is found that only eight of the sixteen subsets are possible. These are indicated in Figure 17.

Example: Flow Matrix Reduction

The flow matrix of Figure 8 can be reduced with the aid of the merger-restriction table of Figure 12, the restriction-dependency table of Figure 16, and the reduction rules. The reduction procedure as outlined in this chapter will be followed.

According to step 4, the subset identified as 0101 is selected from Figure 17. This subset is $\{(1,4),\ (2.6)\}$, and from the restriction-dependency table of Figure 16, it is seen that (2,6) has a restriction of (1,4). Since (1,4) is already a member of the subset, it does not have to be added, and the subset is complete for examination. All combinations are possible, and the diagram of Figure 19 results.

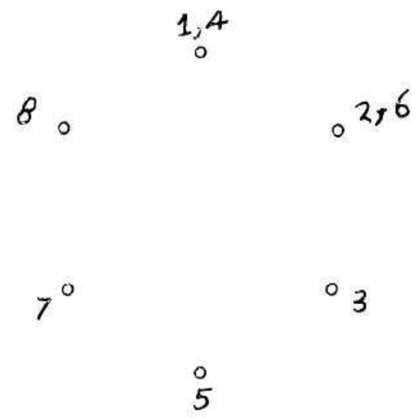

Figure 19. Partial Merger Diagram for Examination of Flow Matrix of Figure 8

Each possible merger of Figure 12 is now examined against the merger diagram of Figure 20 as follows:

(1) Possible merger (1,2) may be accepted if, according to rule 1, (1,6), (2,4) and (4,6) may be merged. Since (1,6) is not a possible merger, according to the table of Figure 12, possible merger (1,2) is rejected.

(2) Possible merger (1,4) has already been combined.

(3) Possible merger (1,5) is not considered since its restriction has not been satisfied.

(4) Possible merger (2,3) satisfies the rules if (3,6) is possible. Since (3,6) is not possible according to the table of Figure 12, (2,3) is rejected.

(5) Possible merger (3,5) fails since its restriction (5,6) has not been combined.

(6) Possible merger (2,6) has already been satisfied as a combination.

(7) Possible merger (2,7) fails since (6,7) is not a possible merger.

(8) Possible merger (2,6) fails since (6,8) is not a possible merger.

(9) Possible merger (3,4) fails since (1,3) is not a possible merger.

(10) Possible merger (3,8) satisfies all rules and is indicated as a possible merger on the merger diagram of Figure 20.

(11) Possible merger (4,5) fails since (1,5) is not a possible merger.

(12) Possible merger (4,7) fails since (1,7) is not a possible merger.

(13) Possible merger (6,8) fails since (2,8) is not a possible merger.

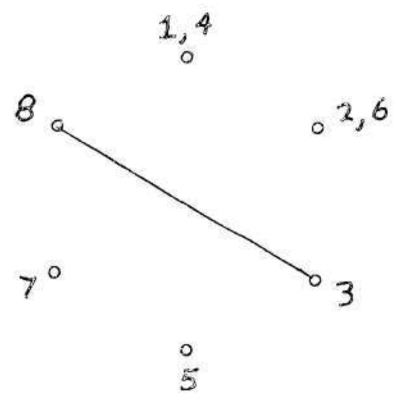

Figure 20. Merger Diagram for the Flow Matrix of Figure 8, based upon the Partial Diagram of Figure 19

From Figure 20, the minimum flow matrix for the subset of restrictions under consideration is a five-state matrix consisting of states $\{1,4\}$, $\{2,6\}$, $\{3,8\}$, $\{5\}$, and $\{7\}$ of the original flow matrix of Figure 8.

The theoretical minimum matrix for the example of Figure 8 was determined to be four states from the examination of Chapter III; therefore, the subset considered above does not yield the theoretical minimum matrix. It is now necessary to examine another subset of restrictions.

Of the remaining subsets indicated in Figure 17, only seven more subsets will yield a reduced flow matrix; however, none of the remaining reduced flow matrices have the theoretical minimum of four states. The conclusion is that the minimum reduced flow matrix for the example of Figure 8 is a five state matrix. (The minimum matrix which was found.) The reduced flow matrix is constructed in Figure 21. The original state numbers states are indicated in Figure 21.

Original State No.	Inputs X_1	X_2	X_3	State
$\{1,4\}$	2(1)	2(1)	3(2)	1
$\{2,6\}$	-	1(2)	5(1)	2
$\{3,8\}$	5(2)	-	2(1)	3
$\{5\}$	4(1)	-	3(2)	4
$\{7\}$	2(2)	-	1(2)	5

Figure 21. The Minimum Reduced Flow Matrix for the Flow Matrix of Figure 8

The reduction procedure as outlined in the previous sections is a systematic procedure for reducing a flow matrox, and in its present form is readily programmed for computer application. A further example of reducing a flow matrix is given in Appendix I, and the computer application of the procedure is discussed in the next chapter.

CHAPTER VI

COMPUTER PROGRAM FOR MATRIX REDUCTION

Introduction

A program based upon the reduction procedure of Chapter IV, has been written for the automatic reduction of a flow matrix. This program selects possible mergers from the merger restriction table, and accepts those mergers for which the restrictions may be satisfied. The program and any necessary modifications required to reduce a flow matrix according to the methods of Chapter V is now presented.

The complete program consists of five segments and one main or control program. Each part is coded in the language of the Burroughs Algebraic Compiler, a dialect of ALGOL,* and each part performs a complete operation according to the reduction procedure.

An outline of the parts of the program is as follows:

(1) Master Program: Declares all variables and arrays, declares the output formats, and directs the overlay of the segments.

(2) Segment Part 1: Constructs the merger-restriction table and determines the theoretical minimum number of states in the reduced flow matrix for Huffman's flow matrix.

---

*A complete description of this language exists (15), including a summary of equivalency between its elements and the elements of the international algebraic language ALGOL. Hereafter, a statement in the language of the Burroughs Algebraic Compiler will be referred to simply as an ALGOL statement.

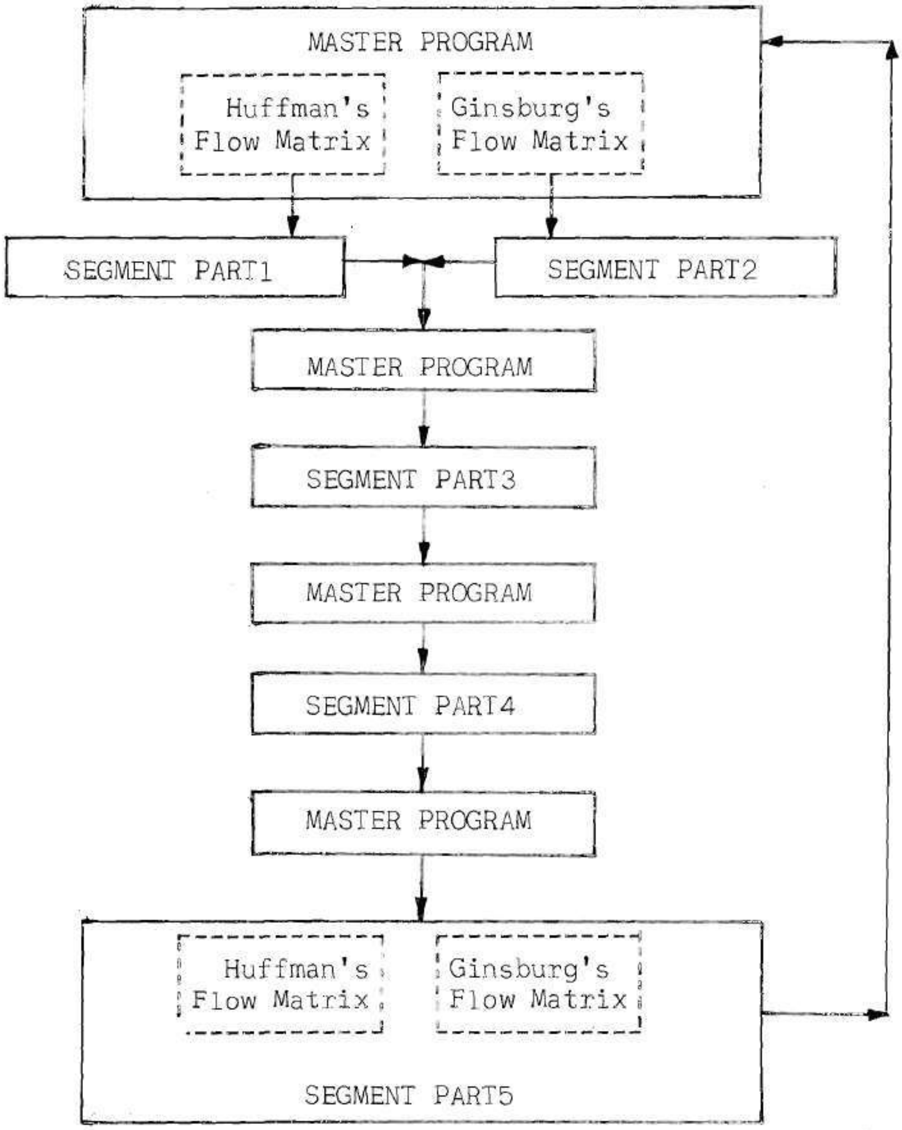

Figure 22. Segmented Block Diagram of Computer Reduction Program

(3) Segment Part 2: Constructs the merger-restriction table and determines the theoretical minimum number of states in the reduced flow matrix for Ginsburg's flow matrix.

(4) Segment Part 3: Constructs the modified merger diagram according to the procedure described in Chapter IV.

(5) Segment Part 4: Determines the optimum mergers from the modified merger diagram as constructed in segment part 3.

(6) Segment Part 5: Constructs the reduced flow matrix as directed by segment part 4.

A segmented block diagram of the program is shown in Figure 22. It is noted that the construction of the merger diagram and the reconstruction of the reduced flow matrices must be different programs according to whether the flow matrix is of Huffman's or Ginsburg's type. However, segments part 3 and part 4 are the same for all types of flow matrices for which the merger-restriction diagram may be constructed.

Master Program and Input Formats

The master program declares all variables and arrays required within the program, along with all printout formats for the output data. All data and mathematical operations within the program must be in either integer or Boolean values; therefore, all input data must be integers.

The complete data required for a single flow matrix will be called a Data Set and must be arranged according to the following format:

(1) The first card of a Data set will be a Boolean code, HUF, indicating whether the flow matrix is of Huffman's class (HUF = 1), or of Ginsburg's class (HUF = 2).

(2) The second card of the Data Set will contain two integers which specify the number of rows (RA1) and the number of columns (CA1) contained in the flow matrix. For Huffman's flow matrix the number of columns will be one greater than the number of allowable inputs. For Ginsburg's flow matrix the number of columns will be the number of allowable inputs. The number of rows will be the number of rows of data contained in the flow matrix.

(3) For Huffman's matrices, the data will be entered in a row by row listing, and the internal state numbers and the output designations will be the same integers as are listed in the flow matrix. To distinguish between internal stable states and internal unstable states, the stable state number must be multiplied by 1000 in the Data Set. A blank internal state in the flow matrix is specified by a zero entry in the Data Set. The output data must be listed as a single Boolean expression and the integer 3 is used in the program to distinguish between a zero (Boolean) output and an unspecified output.

(4) For Ginsburg's matrices, the data will be entered in a row by row listing, alternately listing the next state and next output number of each input column. A blank entry (either next state or next output) must be specified by a zero entry, and for this reason: the different outputs must be specified by integers which are greater than zero.

A five is required in the first column of each card of a Data Set, and a space is required between each different data entry. The program reads only the data required for the flow matrix currently being reduced (i.e., one Data Set) and the different classes of matrices may be arranged

```
5        0        $  COMMENT          GINSBURGS  FLOW  MATRIX  $
5        8           3   $   COMMENT     8 ROWS,    3 COLUMNS  $
5        2           1        0          0          3          2
5        0           0        4          2          0          0
5        7           2        0          0          6          1
5        0           0        2          1          0          0
5        5           1        0          0          3          2
5        0           0        1          2          7          1
5        6           2        0          0          1          2
5        0           0        0          0          2          1
5        1        $  COMMENT          HUFFMANS FLOW MATRIX     $
5        8           5   $   COMMENT     8 ROWS,    5 COLUMNS  $
5     1000           3        0          7          0
5     2000           3        0          7          0
5        1        3000        6          0          1
5        2        4000        5          0          0
5        0           4     5000          8          0
5        0           4     6000          8          0
5        2           0        5       7000          1
5        1           0        6       8000          0
5     SENTINEL         $     COMMENT SENTINEL CARD FOLLOWS THE
5                            LAST DATA SET   $
```

Figure 23. Illustration of the Format Required for Input Data for Matrix Reduction Program

according to any desired order. A SENTINEL card must follow the last Data Set whether one or more matrices are being reduced.

Following the above formats, the input data for the flow matrices of Figures 8 and 10 will be as shown in Figure 23.

THE ALGOL statements of the Master Program are shown in Figure 24. Comment statements are included for the benefit of the reader. It should be pointed out that these comments are not a necessary part of the program, but are included only to make the program itself self-explanatory.

The only input data read by the Master Program is HUF. The rest of the Data Set is read by the proper segments as required within the program.

```
2COMMENT 1      ****************************************************
2        THIS PART OF THE PROGRAM IS DESIGNATED AS THE MASTER PROGRAM.
2        THE STATEMENTS IMMEDIATELY FOLLOWING THIS COMMENT ARE THE
2        DECLARATIONS OF ALL IDENTIFIERS AND DIMENSIONS OF THE ARRAYS
2        USED IN THE PROGRAM. THE PROGRAM BEGINS WITH COMMENT 4.
2               ****************************************************     $
2INTEGER I,I1,I2,I3                                                      $
2INTEGER K,K1,K2,K3,K4,L,L1,L2,L3,L4                                     $
2INTEGER C1,C2,C3,C4,C5,C6,C7,C8,T1,T2,T3,T4,T5,T6,R1,R2,R3,R4,R5        $
2INTEGER CTR,CTR1                                                        $
2INTEGER MN,N,N1,X,Y,X1,X2,B,B1,B2,C                                     $
2INTEGER A1(,),RA1,CA1,A10(,)                                            $
2INTEGER A2(,),RA2                                                       $
2INTEGER A4(),CA4,A6(),CA6,A7(),CA7                                      $
2INTEGER A5(,),RA5,RA5A,RA5B                                             $
2INTEGER A9(,),RA9,CA9                                                   $
2BOOLEAN ALL,HUF,NO,COMB,NO1                                             $
2COMMENT 2      ****************************************************
2        THE FOLLOWING DECLARATION GIVES THE SIZES OF ARRAYS REQUIRED
2        FOR THIS PROGRAM. IN SOME CASES IT MAY BE NECESSARY TO CHANGE
2        THE SIZE OF SOME OR ALL OF THE MATRICES IN ORDER TO HAVE THE
2        PROPER SPACE REQUIRED TO SUCCESSFULLY REDUCE A FLOW MATRIX.
2        THE REQUIRED SIZES MAY BE JUDGED IN MOST CASES AND THE
2        DECLARATION OF SIZES CHANGED AS REQUIRED. IN SOME CASES IT WILL
2        BE IMPOSSIBLE TO SUCCESSFULLY REDUCE A FLOW MATRIX WITH THIS
2        PROGRAM BECAUSE OF THE LACK OF STORAGE SPACE. THE SIZE OF
2        ARRAY SPECIFIED IN EACH CASE IS A LOWER BOUND.
2               ****************************************************     $
2ARRAY   A1(30,5)  $ COMMENT  THIS ARRAY IS USED TO STORE HUFFMANS AND
2                             GINSBURGS FLOW MATRICES. THE ARRAY WILL
2                             REQUIRE AS MANY COLUMNS AND ROWS AS IS
2                             SPECIFIED BY CA1 AND RA1, RESPECTIVELY.    $
2ARRAY   A10(30,4) $ COMMENT  THIS ARRAY STORES THE OUTPUTS ASSOCIATED
2                             WITH GINSBURGS FLOW MATRICES. IT MUST
```

Figure 24. ALGOL Program of Master Program.
(Continued on the next 4 pages.)

```
2                                  CONTAIN AS MANY COLUMNS AND ROWS AS IS
2                                  SPECIFIED BY CA1 AND RA1 RESPECTIVELY.    $
2ARRAY     A5(30,30) $
2ARRAY     A2(30,30) $ COMMENT MATRICES A2(,) AND A5(,) ARE SQUARE
2                                  MATRICES USED IN SEVERAL CAPACITIES
2                                  THROUGHOUT THE PROGRAM. THEY MUST CONTAIN
2                                  AS MANY ROWS AND COLUMNS AS IS SPECIFIED
2                                  BY RA1.                                   $
2ARRAY     A4(15)    $
2ARRAY     A6(15)    $
2ARRAY     A7(15)    $ COMMENT     MATRICES A4(), A6() AND A7() ARE USED IN
2                                  SEVERAL CAPACITIES THROUGHOUT THE PROGRAM.
2                                  THEY MUST CONTAIN AS MANY ROWS AND COLUMNS
2                                  AS IS SPECIFIED BY RA1.                   $
2ARRAY     A9(200,6) $ COMMENT     THE POSSIBLE MERGERS AND THEIR ASSOCIATED
2                                  RESTRICTIONS ARE STORED IN THIS MATRIX.
2                                  THE NUMBER OF COLUMNS REQUIRED FOR THIS
2                                  MATRIX WILL BE ONE GREATER THAN THE NUMBER
2                                  OF ALLOWABLE INPUTS TO THE FLOW MATRIX.
2                                  THE NUMBER OF ROWS REQUIRED WILL BE
2                                  EQUAL TO THE NUMBER OF POSSIBLE MERGERS
2                                  DETERMINED FOR THE FLOW MATRIX. IN SOME
2                                  CASES,IT WILL BE IMPOSSIBLE TO DETERMINE
2                                  THE EXACT NUMBER OF ROWS REQUIRED FOR
2                                  MATRIX A9(,) BEFORE RUNNING THE PROGRAM.
2                                  IF THE NUMBER OF ROWS WHICH HAVE BEEN
2                                  DECLARED IS  NOT ENOUGH TO STORE ALL OF
2                                  THE RESTRICTION FOR THE FLOW MATRIX
2                                  BEING TESTED, AN ERROR MESSAGE IS PRINTED
2                                  ON THE LINE PRINTER, WHICH DECLARES THE
2                                  THE EXACT NUMBER OF ROWS REQUIRED.        $
2COMMENT 3    ***************************************************
2          THE VARIABLES RA9 AND CA9 MUST AGREE WITH THE NUMBER OF
2          ROWS AND COLUMNS DECLARED IN ARRAY A9(,). THEY WILL BE USED AS
```

```
2          AN ERROR CHECK DURING THE PROGRAM IN CASE THE CAPACITY OF
2          THE MATRIX A9(,) SHOULD BE EXCEEDED. THIS IS ALSO THE
2          STARTING POINT OF EACH REDUCTION PROCEDURE.
2                *****************************************************      $
2PP60..    RA9 = 100                     $             CA9 = 6              $
2COMMENT   4     *******************************************************
2          IT IS NECESSARY TO READ A CODE INTO THE PROGRAM WHICH
2          SPECIFIES WHETHER THE NEXT FLOW MATRIX WILL BE GINSBURGS OR
2          HUFFMANS FLOW TABLE.    IF HUF = 1 (TRUE) THEN THE FLOW
2          MATRIX WILL BE HUFFMANS, OTHERWISE HUF = 0 (FALSE)  AND THE
2          FLOW MATRIX WILL BE GINSBURGS. IF A SENTINEL CARD IS READ, THE
2          PROGRAM STOPS.
2                *****************************************************      $
2          READ($ ALL $ DATA)                                               $
2INPUT     DATA(HUF)                                                        $
2          IF ALL                                                           $
2BEGIN     STOP                          $          GO PP60       END       $
2COMMENT   5     *******************************************************
2          IF  THE FLOW MATRIX IS HUFFMANS, THEN IT IS NECESSARY TO
2          USE SEGMENT PART1 IN ORDER TO CALCULATE THE POSSIBLE MERGERS
2          AND THE RESTRICTIONS.   DETERMINATION OF POSSIBLE MERGERS
2          AND NECESSARY RESTRICTIONS FOR GINSBURGS MATRIX IS PREFORMED
2          UNDER SEGMENT PART2. THE CODE HUF DETERMINES WHICH SEGMENT
2          WILL BE USED.
2                *****************************************************      $
2          IF HUF                                                           $
2BEGIN     OVERLAY PART1                                                    $
2          GO PP47                          END                             $
2          OVERLAY PART2                                                    $
2          GO PP61                                                          $
2COMMENT   6     *******************************************************
2          THE PROGRAM WILL RETURN TO THIS POINT AFTER COMPLETION OF
2          EITHER SEGMENT  PART1 OR  PART2, AS THE NEXT TWO SEGMENTS ARE
2          USED REGARDLESS OF THE TYPE OF FLOW MATRIX BEING REDUCED.
```

```
2         SEGMENT PART3 WILL DETERMINE THE FORM OF THE MODIFIED MERGER
2         DIAGRAM WHILE SEGMENT PART4 WILL PERFORM THE MERGER OF THE
2         MODIFIED MERGER DIAGRAM IN AN OPTIMUM MANNER.
2              ****************************************************          $
2PP53..   OVERLAY PART3                                                      $
2         GO PP125                                                           $
2PP71..   OVERLAY PART4                                                      $
2         GO PP72                                                            $
2COMMENT  7    *****************************************************
2         SEGMENT PART5 WILL CONSTUCT THE REDUCED FLOW MATRIX FROM THE
2         MERGERS DETERMINED IN PART4
2              ****************************************************          $
2PP76..   OVERLAY PART5                                                      $
2         GO PP77                                                            $
2COMMENT  8    *****************************************************
2         THE FOLLOWING PROCEDURE WILL ALLOW THE PRINTOUT OF A TWO
2         DIMENSIONAL MATRIX ACCORDING TO THE DESIRED FORMAT. THE
2         VARIABLES ARE,
2            R - THE NUMBER OF ROWS TO BE PRINTED
2            C - THE NUMBER OF COLUMNS TO BE PRINTED
2            A(,) - THE MATRIX TO BE PRINTED
2            N - THE CODE WHICH SPECIFIES THE FORMAT
2              ****************************************************          $
2PROCEDURE MATRIXPRINT1(R,C,A(,),N)                                          $
2BEGIN    INTEGER R,C,A(,),I,J,K,N                                           $
2         FOR I = (1,1,R)                                                    $
2BEGIN    J = 1                                                              $
2         UNTIL J GTR C                                                      $
2BEGIN    EITHER IF N EQL 9                                                  $
2         WRITE($$ROW,F9)                                                    $
2         OR IF N EQL 5                                                      $
2         WRITE($$ROW5,F5)                                                   $
2         OR IF N EQL 2                                                      $
2BEGIN    EITHER IF A(I,1) LSS 1000                                          $
```

```
2          WRITE($$ROW2,F2)                                         $
2          OTHERWISE                                                $
2          WRITE($$ROW21,F21)           END                         $
2          OR IF N EQL 10                                           $
2          WRITE($$ROW10,F10)                                       $
2          OTHERWISE                                                $
2          WRITE($$ROW,F)                                           $
2          J = K                        END         END             $
2OUTPUT    ROW( FOR K = (J,1,MIN(J+14,C)) $ A(I,K))                 $
2FORMAT    F9(I7,I11,B6,9I8,W2)                                     $
2OUTPUT    ROW5(I,FOR K = (J,1,MIN(J + 19,C)) $ A(I,K))             $
2FORMAT    F5(I6,B2,20I5,W2)                                        $
2OUTPUT    ROW21(I,I,A(I,1)/1000)                                   $
2FORMAT    F21(I2,B4,*ROW*,I3,*  MUST MERGE WITH ROW*,I3,W2)        $
2OUTPUT    ROW2(I,FOR K = (J,1,MIN(J+19,C)) $ A(I,K))               $
2FORMAT    F2(I2,I11,B6,20I5,W2)                                    $
2FORMAT    F(15I8,W2)                                               $
2OUTPUT    ROW10(FOR J = (1,1,C) $ (A(I,J),A(I,J+C)) )              $
2FORMAT    F10(4(I5,*(*,I2,*)*),W2)                                 $
2          RETURN END MATRIXPRINT1()                                $
```

Segments One and Two: Merger-Restriction Table

After reading the code HUF, the Master Program will direct the overlay of Segment Part 1 if the matrix is of Huffman's class, or Segment Part 2 if the matrix is of Ginsburg's class. The respective segments first read the code words RA1 and CA1, and then the data of the matrix to be reduced.

Each segment examines its respective flow matrix for the possible compatible state-pairs and then determines which possible compatible state-pairs must be eliminated, leaving only the compatible state-pairs or possible mergers. The merger-restriction table is then constructed, and the theoretical minimum number of states in the reduced flow matrix is determined.

The procedures within the segments are as explained, for the respective flow matrices, in Chapters III and IV. The program segments are shown in Figures 25 and 26. Again, comments are added to Segment Part 1 in order to make it self-explanatory. No comments are included in Segment Part 2 since it follows directly from Segment Part 2 except for the rules of examining the flow matrix.

The segments print out the flow matrix being reduced, the theoretical minimum size determined, and the merger-restriction table. The format is explained within the programs, and the reader is referred to Appendix B for actual printout data of the different flow matrices and tables.

```
2SEGMENT PART1                                   $          BEGIN
2COMMENT 1      ********************************************************
2        SEGMENT PART1 BEGINS HERE. THIS SEGMENT IS USED WHEN IT HAS
2        BEEN DETERMINED THAT THE FLOW MATRIX TO BE REDUCED IS HUFFMANS
2        FLOW MATRIX. THE PURPOSE OF THIS SEGMENT IS TO DETERMINE THE
2        POSSIBLE MERGERS AND THE NECESSARY RESTRICTIONS FOR THE FLOW
2        MATRIX, AND TO CALCULATE THE MINIMUM NUMBER OF ROWS WHICH WILL
2        BE POSSIBLE IN THE REDUCED FLOW MATRIX.
2        THE FLOW MATRIX IS NOW READ INTO MATRIX A1(,), AND STORED
2        THERE THROUGHOUT THE REMAINDER OF THE PROGRAM.
2               *******************************************************    $
2PP47..  READ($$DATA1)                                                     $
2INPUT   DATA1(RA1,CA1, FOR I =(1,1,RA1) $ FOR J =(1,1,CA1) $ A1(I,J))    $
2COMMENT 2      ********************************************************
2        THE FLOW MATRIX WHICH IS TO BE REDUCED  AND A TITLE ARE NOW
2        PRINTED ON THE LINE PRINTER.
2               *******************************************************    $
2        WRITE($$TITL1)                                                    $
2FORMAT  TITL1(*MATRIX A1  HUFFMANS FLOW MATRIX*,W3,
2               *    ---------  INPUTS-----------   OUTPUTS*,W0,
2               *    --A--   --B--   --C--   --D--   -------*,W0)          $
2        MATRIXPRINT1(RA1,CA1,A1(,),1)                                     $
2COMMENT 3      ********************************************************
2        MATRIX A2(,) MUST BE CLEARED TO ZERO. THIS MATRIX WILL BE USED
2        TO STORE INFORMATION WHICH INDICATES POSSIBLE MERGERS OF ROWS.
2        ONLY THE FIRST RA1 ROWS AND RA1 COLUMNS OF THE MATRIX A2(,)
2        MUST BE CLEARED.
2               *******************************************************    $
2        FOR I = (1,1,RA1)                                                 $
2BEGIN   FOR J = (1,1,RA1)                                                 $
2        A2(I,J) = 0                      END                              $
2COMMENT 4      ********************************************************
2        ALL POSSIBLE COMPATIBLE PAIRS ARE NOW DETERMINED ACCORDING
2        TO THE REQUIREMENTS  FOR HUFFMANS FLOW MATRIX.      EACH ROW
```

Figure 25. ALGOL Program of Segment Part 1.
(Continued on the next 9 pages.)

```
2          IS COMPARED WITH ALL OTHER ROWS WITH ONLY ONE TEST BEING
2          REQUIRED. IF TWO ROWS DO NOT SIMULTANEOUSLY HAVE STABLE STATES
2          OCCURING WITHIN THE SAME INPUT COLUMN,     OR IF THE
2          OUTPUTS OF THE TWO ROWS ARE NOT THE SAME, THEN THE ROWS ARE
2          POSSIBLE COMPATIBLE STATE PAIRS. IF THE TWO ROWS (HEREIN
2          DESIGNATED AS I AND J) ARE DETERMINED TO BE POSSIBLE COMPATIBLE
2          ROWS, THEN MATRIX A2(,) WILL BE CHANGED FROM A ZERO TO A ONE
2          IN TWO POSITIONS.  THESE POSITIONS ARE A2(I,J), AND A2(J,I).
2          OTHERWISE, MATRIX A2(,) WILL REMAINED UNCHANGED.
2               ***************************************************      $
2          FOR I = (1,1,RA1)                                             $
2BEGIN     FOR J = (I+1,1,RA1)                                           $
2BEGIN     NO = 0                                                        $
2          FOR I1 = (1,1,CA1-1)                                          $
2BEGIN     IF A1(I,I1) GEQ 1000                                          $
2BEGIN     IF A1(J,I1) GEQ 1000                                          $
2BEGIN     IF A1(I,CA1) NEQ A1(J,CA1)                                    $
2BEGIN     NO = 1                                                        $
2          GO PP41             END        END        END        END      $
2          A2(I,J) = A2(J,I) = 1                                         $
2PP41..    IF NO                                                         $
2BEGIN     A2(J,I) = 0                                                   $
2          A2(I,J) = 0                    END        END        END      $
I          FOR I = (1,1,RA1)                                             $
2          A2(I,I) = 1                                                   $
2COMMENT 5      ***************************************************
2          ALL POSSIBLE COMPATIBLE ROWS HAVE BEEN DETERMINED AT THIS TIME.
2          IT IS NOW NECESSARY TO ELIMINATE ANY OF THE POSSIBLE COMPATIBLE
2          ROWS WHICH ARE ACTUALLY INCOMPATIBLE. CTR AND CTR1 ARE USED
2          AS COUNTERS TO DETERMINE WHEN ALL ELIMINATIONS HAVE BEEN
2          COMPLETED.
2          CRT1 RECORDS THE NUMBER OF POSSIBLE MERGERS WHILE A PASS IS
2          MADE TO REMOVE AS MANY INCOMPATIBLE STATES AS POSSIBLE.  THE
2          NUMBER OF PROBABLE  COMPATIBLE ROWS   REMAINING ARE COUNTED
```

```
2          AS CTR AND COMPARED AGAINST CTR1. IF THE DIFFERENCE IS NOT
2          ZERO, THEN CTR1 IS SET EQUAL TO CTR AND ANOTHER PASS IS MADE.
2               **************************************************      $
2          CTR = 0                                                      $
2          FOR I = (1,1,RA1)                                            $
2BEGIN     FOR J = (I+1,1,RA1)                                          $
2BEGIN     IF A2(I,J) EQL 1                                             $
2          CTR = CTR + 1                  END        END                $
2COMMENT 6      ***************************************************
2          THE FIRST COUNT HAS BEEN PERFORMED AND NOW THE PROCESS OF
2          ELIMINATION INVOLVES TESTING ALL POSSIBLE COMPATIBLE ROWS
2          WITH RESPECT TO EVERY POSSIBLE INPUT TO DETERMINE IF,
2          1)EITHER ROW HAS A BLANK ENTRY.
2          2)BOTH ROWS HAVE A STABLE STATE ENTRY.
2          3)NEITHER ROW HAS A STABLE STATE ENTRY,BUT BOTH ROWS WILL
2               ADVANCE TO EITHER THE SAME ROW, OR TO DIFFERENT ROWS WHICH
2               ARE LISTED AS POSSIBLE MERGERS IN MATRIX A2(,).
2          4)ONLY ONE ROW HAS AN ENTRY WHICH IS A STABLE STATE, BUT THE
3               SECOND ROW ADVANCES TO A THIRD ROW SUCH THAT THE FIRST AND
2               THIRD ROW ARE INDICATED AS A POSSIBLE COMPATIBLE STATE
2               PAIR IN MATRIX A2(,).
2          IF ONE OF THE ABOVE TESTS IS NOT PASSED WITH RESPECT TO EVERY
2          INPUT, THEN THE PAIR UNDER EXAMINATION IS REMOVED AS A POSSIBLE
2          COMPATIBLE ROW. THIS IS DONE BY SETTING A2(I,J) AND A2(J,I)
2          EQUAL TO ZERO.
2               **************************************************      $
2PP43..    CTR1 = CTR                                                   $
2          FOR I = (1,1,RA1)                                            $
2BEGIN     FOR J = (I+1,1,RA1)                                          $
2BEGIN     IF A2(I,J) EQL 1                                             $
2BEGIN     FOR I1 = (1,1,CA1-1)                                         $
2BEGIN     K = A1(I,I1)                                                 $
2          K1 = A1(J,I1)                                                $
2          IF (K EQL 0) OR (K1 EQL 0)                                   $
```

```
2          GO PP42                                                           $
2          IF (K GEQ 1000) AND (K1 GEQ 1000)                                 $
2          GO PP42                                                           $
2          EITHER IF (K GEQ 1000) AND (K1 LSS 1000)                          $
2BEGIN     L1 = (K1)(1000)                                                   $
2          FOR I2 = (1,1,RA1)                                                $
2BEGIN     IF A1(I2,I1) EQL L1                                               $
2BEGIN     IF A2(I,I2) EQL 0                                                 $
2          A2(I,J) = A2(J,I) = 0          END       END       END            $
2          OR IF (K LSS 1000) AND (K1 GEQ 1000)                              $
2BEGIN     L1 = (K)(1000)                                                    $
2          FOR I2 = (1,1,RA1)                                                $
2BEGIN     IF A1(I2,I1) EQL L1                                               $
2BEGIN     IF A2(J,I2) EQL 0                                                 $
2          A2(I,J) = A2(J,I) = 0          END       END       END            $
2          OTHERWISE                                                         $
2BEGIN     IF A2(K,K1) EQL 0                                                 $
2          A2(J,I) = A2(I,J) = 0          END                                $
2PP42..                         END       END       END       END            $
2COMMENT 7     ****************************************************
2          A COMPLETE PASS THROUGH MATRIX A2(,) HAS BEEN COMPLETED. A
2          COUNT IS NOW MADE TO DETERMINE IF ANOTHER PASS IS REQUIRED.
2          IF CTR AND CTR1 ARE NOT EQUAL THEN THE ABOVE PROCESS OF
2          ELIMINATION IS REPEATED.
2              ***************************************************           $
2          CTR = 0                                                           $
2          FOR I = (1,1,RA1)                                                 $
2BEGIN     FOR J = (I+1,1,RA1)                                               $
2BEGIN     IF A2(I,J) EQL 1                                                  $
2          CTR = CTR + 1                  END       END                      $
2          IF CTR NEQ CTR1                                                   $
2          GO PP43                                                           $
2COMMENT 8     ****************************************************
2          THE PROCESS OF ELIMINATION HAS BEEN COMPLETED AT THIS POINT.
```

```
2          ALL PAIRS LISTED WITH A 1 IN MATRIX A2(,) ARE COMPATIBLE PAIRS
2          AND ARE THEREFORE POSSIBLE MERGERS. A CHECK IS NOW MADE TO
2          DETERMINE IF MATRIX A9(,) IS LARGE ENOUGH TO CONTAIN A LIST
2          OF ALL POSSIBLE MERGERS.  IF MATRIX A9(,) IS NOT LARGE ENOUGH,
2          THEN A WARNING IS PRINTED ON THE LINE PRINTER AND THE MERGING
2          PROCESS IS CONCLUDED FOR THIS FLOW MATRIX. OTHERWISE, THE
2          REQUIRED SPACE IN MATRIX A9(,) IS CLEARED FOR STORAGE OF ALL
2          POSSIBLE MERGERS.
2               *****************************************************     $
2          K = CTR/2                                                       $
2          IF CTR GTR RA9                                                  $
2BEGIN     WRITE($$ARM1,ALA1)                                              $
2FORMAT    ALA1(*THE NUMBER OF POSSIBLE MERGERS IS*,I5,W2,
2               *THIS EXCEEDS THE AVAILABLE STORAGE*,W0)                   $
2OUTPUT    ARM1(K)                                                         $
2          GO PP60                        END                              $
2          FOR I = (1,1,CTR)                                               $
2BEGIN     FOR J = (1,1,CA9)                                               $
2          A9(I,J) = 0                    END                              $
2COMMENT   9    *****************************************************
2          A LIST IS NOW MADE OF ALL POSSIBLE MERGERS AND THEIR ASSOCIATED
2          RESTRICTIONS. THIS LIST IS STORED IN MATRIX A9(,) IN THE
2          FOLLOWING FORMAT. ALL PAIRS OF ROWS WHICH ARE POSSIBLE MERGERS
2          ARE LISTED IN COLUMN ONE OF MATRIX A9(,). THERE WILL BE AS
2          MANY ROWS REQUIRED AS THERE ARE POSSIBLE MERGERS. THE POSSIBLE
2          MERGER  PAIRS ARE STORED AS ONE NUMBER BY MULTIPLYING THE
2          SMALLEST NUMBER BY 1000 AND ADDING IT TO THE LARGER NUMBER
2          (I.E., (I)(1000)+J). RA9 WILL BE SET EQUAL TO ZERO AND USED TO
2          COUNT THE NUMBER OF POSSIBLE MERGERS.  THE SECOND COLUMN OF THE
2          MATRIX STORES AN INTEGER WHICH RELATES THE NUMBER OF
2          RESTRICTIONS ASSOCIATED WITH THE POSSIBLE MERGER LISTED IN THE
2          FIRST COLUMN OF THAT SAME ROW. THE REMAINING COLUMNS IN EACH
2          ROW STORE  ANY NECESSARY RESTRICTIONS FOR THE POSSIBLE MERGER
2          IN COLUMN ONE OF THE RESPECTIVE ROW. EACH RESTRICTION WILL BE
```

```
2          A PAIR OF ROWS AND WILL BE STORED AS ONE NUMBER BY MULTIPLYING
2          THE SMALLEST NUMBER BY 1000 AND ADDING IT TO THE LARGEST NUMBER
2          OF THE PAIR.  THE MAXIMUM NUMBER OF RESTRICTIONS FOR ANY
2          POSSIBLE MERGER IS EQUAL TO THE NUMBER OF INPUTS.
           THEREFORE THE MAXIMUM NUMBER OF COLUMNS REQUIRED FOR MATRIX
2          A9(,) IS EQUAL TO TWO PLUS THE ALLOWABLE NUMBER OF INPUTS FOR
2          THE FLOW TABLE.
2               ******************************************************    $
2          RA9 = 0                                                          $
2          FOR I = (1,1,RA1)                                                $
2BEGIN     FOR J = (I+1,1,RA1)                                              $
2BEGIN     IF A2(I,J) EQL 1                                                 $
2BEGIN     RA9 = RA9 + 1                                                    $
2          A9(RA9,1) = I(1000) + J                                          $
2COMMENT   10   ******************************************************
2          IT HAS BEEN DETERMINED THAT ROWS I AND J ARE POSSIBLE MERGERS,
2          AND THEY HAVE BEEN STORED IN THE FIRST COLUMN OF ROW RA9 OF
2          MATRIX A9(,). EACH INPUT COLUMN OF THE TWO ROWS WILL NOW BE
2          CHECKED TO DETERMINE ANY RESTRICTIONS. NO RESTRICTIONS WILL
2          EXIST, FOR THE INPUT COLUMN UNDER CONSIDERATION, IF EITHER OF
2          THE FOLLOWING SITUATIONS EXIST,
2          1)EITHER ROW HAS A BLANK ENTRY,
2          2)BOTH ROWS HAVE A STABLE STATE,
2          3)EITHER ROW HAS A STABLE STATE,WHILE THE OTHER ROW HAS AN
2               UNSTABLE STATE OF THE SAME NUMBER AS THE STABLE STATE,
2          4)BOTH ROWS CONTAIN AN UNSTABLE STATE OF THE SAME NUMBER.
2               ******************************************************    $
2          FOR I1 = (1,1,CA1-1)                                             $
2BEGIN     K = A1(I,I1)                                                     $
2          K1 = A1(J,I1)                                                    $
2          IF (K EQL 0) OR (K1 EQL 0)                                       $
2          GO PP44                                                          $
2          IF (K GEQ 1000) AND (K1 GEQ 1000)                                $
2          GO PP44                                                          $
```

```
2           IF K EQL K1                                                      $
2           GO PP44                                                          $
2           IF K/1000 EQL K1                                                 $
2           GO PP44                                                          $
2           IF K1/1000 EQL K                                                 $
2           GO PP44                                                          $
2           IF K EQL K1                                                      $
2           GO PP44                                                          $
2COMMENT    11    *****************************************************
2           THE INPUT UNDER CONSIDERATION HAS NOT BEEN ELIMINATED ACCORDING
2           TO COMMENT 10   , THEREFORE THE RESTRICTION MUST BE
2           DETERMINED AND LISTED IN MATRIX A9(,). A RESTRICTION WILL BE
2           LISTED FOR ROWS I AND J IF,
2           1)ONE ROW IS A STABLE STATE, BUT THE OTHER ROW IS AN UNSTABLE
2                STATE FOR A STABLE STATE IN A THIRD ROW. THE RESTRICTION
2                WILL BE THE TWO ROWS WHICH CONTAIN THE STABLE STATES.
2           2)BOTH ROWS CONTAIN UNSTABLE STATE ENTRIES.  THE RESTRICTION
2                WILL BE THE TWO ROWS WHICH CONTAIN THE RESPECTIVE
2                STABLE STATES.
2                *****************************************************  $
2           L = A9(RA9,2) + 3                                                $
2           A9(RA9,2) = A9(RA9,2) + 1                                        $
2           IF (K GEQ 1000) AND (K1 LSS 1000)                                $
2BEGIN      L1 = (K1)(1000)                                                  $
2           FOR I2 = (1,1,RA1)                                               $
2BEGIN      IF A1(I2,I1) EQL L1                                              $
2BEGIN      EITHER IF I LSS I2                                               $
2           A9(RA9,L) = I(1000) + I2                                         $
2           OTHERWISE                                                        $
2           A9(RA9,L) = (I2)1000 + I                                         $
2           GO PP44                      END         END         END         $
2           IF (K LSS 1000) AND (K1 GEQ 1000)                                $
2BEGIN      L1 = K(1000)                                                     $
2           FOR I2 = (1,1,RA1)                                               $
```

```
2BEGIN    IF A1(I2,I1) EQL L1                                                    $
2BEGIN    EITHER IF J LSS I2                                                     $
2         A9(RA9,L) = (J)1000 + I2                                               $
2         OTHERWISE                                                              $
2         A9(RA9,L) = I2(1000) + J                                               $
2         GO PP44                        END        END        END               $
2         K = K(1000)                                                            $
2         K1 = K1(1000)                                                          $
2         FOR K2 = (1,1,RA1)                                                     $
2BEGIN    IF A1(K2,I1) EQL K                                                     $
2BEGIN    K = K2                                                                 $
2         GO PP45                        END        END                          $
2PP45..   FOR K2 = (1,1,RA1)                                                     $
2BEGIN    IF A1(K2,I1) EQL K1                                                    $
2BEGIN    K1 = K2                                                                $
2         GO PP46                        END        END                          $
2PP46..   EITHER IF K1 LSS K                                                     $
2         A9(RA9,L) = K1(1000) + K                                               $
2         OTHERWISE                                                              $
2         A9(RA9,L) = K(1000) + K1                                               $
2COMMENT 12     *******************************************************
2         AT THIS POINT, TWO ROWS, WHICH ARE POSSIBLE MERGERS, HAVE BEEN
2         EXAMINED WITH RESPECT TO ONE INPUT TO DETERMINE IF A
2         RESTRICTION EXISTS. IF SO, THE RESTRICTION WAS RECORDED. THIS
2         PROCEDURE IS REPEATED AS MANY TIMES AS IS REQUIRED.
2               ******************************************************   $
2PP44..                       END        END        END        END               $
2COMMENT 13     *******************************************************
2         ALL RESTRICTIONS HAVE NOW BEEN DETERMINED. BEFORE DESTROYING
2         MATRIX A2(,), IT IS NECESSARY TO CALCULATE THE MINIMUM
2         NUMBER OF ROWS WHICH MAY BE OBTAINED IN A REDUCED FLOW
2         MATRIX. THE MINIMUM NUMBER WILL BE RECORDED AS THE VARIABLE MN.
2               ******************************************************   $
2         MN = 0                                                                 $
```

```
2         FOR I = (1,1,RA1)                                                       $
2BEGIN    RA6 = RA4 = 0                                                           $
2         FOR J = (I+1,1,RA1)                                                     $
2BEGIN    IF A2(I,J) EQL 0                                                        $
2BEGIN    RA4 = RA4 + 1                                                           $
2         A4(RA4) = J                         END       END                       $
2         FOR I1 = (1,1,RA4)                                                      $
2BEGIN    K = A4(I1)                                                              $
2         FOR J = (1,1,RA6)                                                       $
2BEGIN    K1 = A6(J)                                                              $
2         IF A2(K,K1) EQL 1                                                       $
2         GO PP49                             END                                 $
2         RA6 = RA6 + 1                                                           $
2         A6(RA6) = K                                                             $
2PP49..                                       END                                 $
2         IF MN LSS RA6 + 1                                                       $
2BEGIN    MN = RA6 + 1                                                            $
2         RA7 = 1                                                                 $
2         FOR I1 = (1,1,RA1)                                                      $
2         A7(I1) = 0                                                              $
2         A7(1) = I                                                               $
2         FOR I1 = (1,1,RA6)                                                      $
2BEGIN    RA7 = RA7 + 1                                                           $
2         A7(RA7) = A6(I1)                    END       END       END             $
2         WRITE($$NUM,MI)                                                         $
2FORMAT   MI(*MINIMUM NUMBER OF ROWS ATTAINABLE -*,I3,W4,
2            *THE ROW NUMBERS ARE*,15I4,W0)                                       $
2OUTPUT   NUM(MN, FOR I = (1,1,MN) $ A7(I))                                       $
2COMMENT  14    *****************************************************
2         THE POSSIBLE MERGERS AND ANY ASSOCIATED RESTRICTIONS ALONG
2         WITH A TITLE ARE NOW PRINTED ON THE LINE PRINTER ACCORDING TO
2         THE NEXT COMMANDS.
2               ****************************************************      $
2         WRITE($$TITL9)                                                          $
```

```
2FORMAT   TITL9(*MATRIX A9-RESTRICTION MATRIX*,W4,
2               *POSSIBLE    NUMBER  OF*,WO,
2               * MERGERS   RESTRICTIONS    ----RESTRICTIONS----*,WO)  $
2         MATRIXPRINT1(RA9,CA9,A9(,),9)                                $
2COMMENT  15    ****************************************************
2         THIS CONCLUDES SEGMENT PART1. THE REQUIRED OUTPUT OF THIS
2         SEGMENT IS,
2              RA9 - THE NUMBER OF POSSIBLE MERGERS.
2              A9(,) - LISTING OF POSSIBLE MERGERS AND ANY NECESSARY
2                      RESTRICTIONS.
2              MN - THE SIZE OF THE MINIMUN REDUCED FLOW MATRIX
2                   ATTAINABLE.
2         THE PROGRAM NOW JUMPS TO SEGMENT PART3  VIA THE MASTER PROGRAM.
2              ****************************************************    $
2         GO PP53                                                      $
2         END PART1                                                    $
```

```
2SEGMENT PART2                                $          BEGIN
2INPUT   DATA10(RA1,CA1, FOR I=(1,1,RA1) $ FOR J =(1,1,CA1) $ (A1(I,J),
2                   A10(I,J)))                                                    $
2FORMAT  TITL1G(*MATRIX A1  GINSBURG STATE IDENIFICATION*,W3,
2              *    --------- INPUTS------------*,WO,
2              *    --A--    --B--    --C--    --D--*,WO)                         $
2FORMAT  TITL10(*MATRIX A10   GINSBURG OUTPUT IDENIFICATION*,W4,
2              *    ---------- OUTPUTS-----------*,WO,
2              *    --------FOR INPUTS-----------*,WO,
2              *    --A--    --B--    --C--    --D--*,WO)                         $
2PP61..  READ($$DATA10)                                                           $
2        WRITE($$TITL1G)                                                          $
2        MATRIXPRINT1(RA1,CA1,A1(,),1)                                            $
2        WRITE($$TITL10)                                                          $
2        MATRIXPRINT1(RA1,CA1,A10(,),1)                                           $
2        FOR I = (1,1,RA1)                                                        $
2BEGIN   FOR J = (1,1,RA1)                                                        $
2        A2(I,J) = 0                        END                                   $
2        FOR I = (1,1,RA1)                                                        $
2BEGIN   FOR J = (I+1,1,RA1)                                                      $
2BEGIN   CTR = 0                                                                  $
2        FOR I1 = (1,1,CA1)                                                       $
2BEGIN   EITHER IF A10(I,I1) EQL 0                                                $
2        CTR = CTR + 1                                                            $
2        OR IF A10(J,I1) EQL 0                                                    $
2        CTR = CTR + 1                                                            $
2        OR IF A10(I,I1) EQL A10(J,I1)                                            $
2        CTR = CTR + 1                      END         END                       $
2        IF CTR EQL CA1                                                           $
2BEGIN   A2(I,J) = 1                                                              $
2        A2(J,I) = 1                        END         END         END           $
2        FOR I = (1,1,RA1)                                                        $
2        A2(I,I) = 1                                                              $
2        CTR = 0                                                                  $
```

Figure 26. ALGOL Program of Segment Part 2.
(Continued on the next 3 pages.)

```
2              FOR I = (1,1,RA1)                                                     $
2BEGIN         FOR J = (I+1,1,RA1)                                                   $
2BEGIN         IF A2(I,J) EQL 1                                                      $
2              CTR = CTR + 1                       END        END                    $
2PP63..        CTR1 = CTR                                                            $
2              FOR I = (1,1,RA1)                                                     $
2BEGIN         FOR J = (I+1,1,RA1)                                                   $
2BEGIN         IF A2(I,J) EQL 1                                                      $
2BEGIN         FOR I1 = (1,1,CA1)                                                    $
2BEGIN         K = A1(I,I1)                                                          $
2              K1 = A1(J,I1)                                                         $
2              IF (K EQL 0) OR (K1 EQL 0)                                            $
2              GO PP62                                                               $
2              IF A2(K,K1) EQL 0                                                     $
2BEGIN         A2(I,J) = 0                                                           $
2              A2(J,I) = 0                         END                               $
2PP62..                             END        END        END        END             $
2              CTR = 0                                                               $
2              FOR I = (1,1,RA1)                                                     $
2BEGIN         FOR J = (I+1,1,RA1)                                                   $
2BEGIN         IF A2(I,J) EQL 1                                                      $
2              CTR = CTR + 1                       END        END                    $
2              IF CTR NEQ CTR1                                                       $
2              GO PP63                                                               $
2              IF CTR GTR RA9                                                        $
2BEGIN         MATRIXPRINT1(RA1,RA1,A2(,),1)                                         $
2              GO PP60                             END                               $
2              FOR I = (1,1,CTR)                                                     $
2BEGIN         FOR J = (1,1,CA9)                                                     $
2              A9(I,J) = 0                         END                               $
2              RA9 = 0                                                               $
2              FOR I = (1,1,RA1)                                                     $
2BEGIN         FOR J = (I+1,1,RA1)                                                   $
2BEGIN         IF A2(I,J) EQL 1                                                      $
```

```
2BEGIN    RA9 = RA9 + 1                                                    $
2         A9(RA9,1) = I(1000) + J                                          $
2         FOR I1 = (1,1,CA1)                                               $
2BEGIN    K = A1(I,I1)                                                     $
2         K1 = A1(J,I1)                                                    $
2         IF (K EQL 0) OR (K1 EQL 0)                                       $
2         GO PP64                                                          $
2         IF K EQL K1                                                      $
2         GO PP64                                                          $
2         EITHER IF K1 LSS K                                               $
2         K2 = (K1)(1000) + K                                              $
2         OTHERWISE                                                        $
2         K2 = (K)(1000) + K1                                              $
2         IF K2 EQL A9(RA9,1)                                              $
2         GO PP64                                                          $
2         L = A9(RA9,2) + 3                                                $
2         A9(RA9,2) = A9(RA9,2) + 1                                        $
2         A9(RA9,L) = K2                                                   $
2PP64..                         END       END       END       END          $
2         MN = 0                                                           $
2         FOR I = (1,1,RA1)                                                $
2BEGIN    RA6 = RA4 = 0                                                    $
2         FOR J = (I+1,1,RA1)                                              $
2BEGIN    IF A2(I,J) EQL 0                                                 $
2BEGIN    RA4 = RA4 + 1                                                    $
2         A4(RA4) = J                     END       END                    $
2         FOR I1 = (1,1,RA4)                                               $
2BEGIN    K = A4(I1)                                                       $
2         FOR J = (1,1,RA6)                                                $
2BEGIN    K1 = A6(J)                                                       $
2         IF A2(K,K1) EQL 1                                                $
2         GO PP69                         END                              $
2         RA6 = RA6 + 1                                                    $
2         A6(RA6) = K                                                      $
```

```
2PP69..                                              END                                       $
2          IF MN LSS RA6 + 1                                                                   $
2BEGIN     MN = RA6 + 1                                                                        $
2          RA7 = 1                                                                             $
2          FOR I1 = (1,1,15)                                                                   $
2          A7(I1) = 0                                                                          $
2          A7(1) = I                                                                           $
2          FOR I1 = (1,1,RA6)                                                                  $
2BEGIN     RA7 = RA7 + 1                                                                       $
2          A7(RA7) = A6(I1)                          END        END        END                 $
2          WRITE($$NUM1,MI1)                                                                   $
2OUTPUT    NUM1(MN, FOR I = (1,1,MN) $ A7(I))                                                  $
2FORMAT    MI1(*MINIMUM NUMBER OF ROWS ATTAINABLE -*,I3,W4,
2              *THE ROW NUMBERS ARE*,15I4,W0)                                                  $
2          WRITE($$TITL90)                                                                     $
2FORMAT    TITL90(*MATRIX A9-RESTRICTION MATRIX*,W4,
2                *POSSIBLE     NUMBER  OF*,W0,
2                * MERGERS   RESTRICTIONS    -----RESTRICTIONS----*,W0)                        $
2          MATRIXPRINT1(RA9,CA9,A9(,),9)                                                       $
2          GO PP53                                                                             $
2          END PART2                                                                           $
```

Segment Three: Construction of Merger Diagram

Segment Part 3 constructs the merger diagram from the merger-restriction table as provided by either of the two previous segments. The input data required by this segment is the data as explained by comment 15 of Figure 25. The output data of this segment is explained in comment 37 of Figure 27. An example of the format of the merger diagram is shown in Appendix B.

This segment of the program uses several subroutines for tests which are repeated, and again comments have been added to the program to explain the tests being conducted. The construction of the merger diagram follows the same procedure outlined in Chapter IV.

```
2SEGMENT PART3                              $          BEGIN
2COMMENT 1     ******************************************************
2        SEGMENT PART3 BEGINS HERE.
2        THIS SEGMENT OF THE PROGRAM WORKS WITH THE MATRIX A9(,) AS
2        PROVIDED BY EITHER SEGMENT PART1 OR SEGMENT PART2, TO PRODUCE
2        THE MODIFIED MERGER DIAGRAM. THE CONSTRUCTION OF THE DIAGRAM
2        WILL BE THROUGH A TRIAL PROCEDURE WHICH IS CONTROLLED BY THE
2        PROGRAM. A TRIAL WILL BE THE COMPLETE CONSIDERATION OF A
2        POSSIBLE MERGER AS INDICATED IN MATRIX A9(,), ALONG WITH ALL
2        ASSOCIATED RESTRICTIONS. ASSOCIATED RESTRICTIONS INCLUDE NOT
2        ONLY THE RESTRICTIONS RELATED DIRECTLY TO A POSSIBLE MERGER,
2        BUT ALSO RESTRICTIONS WHICH EXIST UPON THE DIRECT
2        RESTRICTIONS.  A TRIAL WILL BEGIN BY ASSUMING THAT A
2        POSSIBLE MERGER LISTED IN MATRIX A9(,) IS ACTUALLY POSSIBLE.
2        THIS POSSIBLE MERGER IS THEN INDICATED IN THE MODIFIED MERGER
2        DIAGRAM AND ALL ASSOCIATED RESTRICTIONS ARE THEN TESTED TO
2        SEE IF THEY MAY BE FULFILLED. AS EACH RESTRICTION IS TESTED
2        AND IT IS DETERMINED THAT THE RESTRICTION HAS PASSED, THE
2        PROPER COMBINATION IS MADE IN THE MODIFIED MERGER DIAGRAM.
2        A RESTRICTION IS SATISFIED ONLY WHEN THE TWO ROWS WHICH
2        COMPOSE THE RESTRICTION MAY BE COMBINED IN THE MODIFIED MERGER
2        DIAGRAM.
2        IF ANY TEST IS UNSUCCESSFUL, THE TRIAL IS TERMINATED AND THE
2        MERGER DIAGRAM IS CORRECTED TO ITS PREVIOUS CONDITION. SHOULD
2        ALL RESTRICTIONS BE SATISFIED, THEN THE MODIFIED MERGER DIAGRAM
2        IS ACCEPTED AND A NEW TRIAL IS BEGUN.
2        HEREAFTER, THE MODIFIED MERGER DIAGRAM SHALL BE KNOWN AS THE
2        MERGER DIAGRAM.
2              *****************************************************     $
2SUBROUTINE CHCKMERG                                                      $
2COMMENT 2     ******************************************************
2        THIS SUBROUTINE CHECKS TWO GIVEN ROW NUMBERS (T5 AND T6)
2        AGAINST THE MATRIX A9(,) TO DETERMINE IF THE PAIR FORMS A
2        POSSIBLE MERGER THAT IS LISTED IN MATRIX A9(,). IF SO, THEN THE
```

Figure 27. ALGOL Program of Segment Part 3.
(Continued on the next 16 pages.)

```
2         ROW NUMBER IN MATRIX A9(,) IS INDICATED AS T1, OTHERWISE THE
2         BOOLEAN OPERATOR NO IS SET EQUAL TO 1 TO INDICATE THAT A MERGER
2         IS IMPOSSIBLE.
2              ***************************************************          $
2BEGIN    EITHER IF T5 LSS T6                                               $
2         T3 = (T5)(1000) + T6                                              $
2         OTHERWISE                                                         $
2         T3 = (T6)(1000) + T5                                              $
2         FOR T5 = (1,1,RA9)                                                $
2BEGIN    IF A9(T5,1) EQL T3                                                $
2BEGIN    T1 = T5                                                           $
2         GO TP2                              END         END               $
2         NO = 1                                                            $
2TP2..    RETURN END CHCKMERG                                               $
2SUBROUTINE CHCKCOMB                                                        $
2COMMENT 3     ****************************************************
2         THIS SUBROUTINE PERFORMS TWO DIFFERENT TESTS ACCORDING TO
2         A BOOLEAN CODE COMB.  IF COMB EQUALS 1, THEM THE TWO ROWS
2         INDICATED AS C1 AND C2 ARE TESTED AS ROWS WHICH ARE TO
2         BE COMBINED. OTHERWISE, COMB EQUALS 0, AND THE ROWS ARE TESTED
2         AS ROWS WHICH ARE TO TO INDICATED AS MERGERS IN THE
2         MERGER DIAGRAM.
2         A SUCCESSFUL TEST MUST FULFILL THE FOLLOWING REQUIREMENTS FOR
2         ALL MERGERS AND COMBINATIONS INCLUDED WITHIN THE TEST,
2         1)IF TWO ROWS ARE TO BE MERGED, THEN EACH ROW MUST BE ABLE
2             TO BE MERGED WITH ANY ROW(S) WHICH HAS PREVIOUSLY BEEN
2             COMBINED WITH THE OTHER ROW.
2          2) IF TWO ROWS ARE TO BE COMBINED, THEN EACH ROW MUST BE ABLE
2             TO COMBINE WITH ANY ROW(S) WHICH HAS PREVIOUSLY BEEN
2             COMBINED WITH THE OTHER ROW.
2         3)IF TWO ROWS ARE TO BE COMBINED, IT IS NECESSARY THAT EACH ROW
2             BE ABLE TO MERGE WITH ANY ROW(S) WHICH HAS  PREVIOUSLY BEEN
2             MERGED WITH THE OTHER ROW.
2         TEST 1 IS PERFORMED BY PART 1 OF THIS SUBROUTINE  WHILE TESTS
```

```
2          2 AND 3 ARE PERFORMED BY PART 2 OF THIS SUBROUTINE.
2          THE TEST CONSISTS OF DETERMINING THE NECESSARY ADDITIONAL
2          MERGERS AND COMBINATIONS AND LISTING ANY REQUIRED RESTRICTIONS
2          IN MATRIX A4().
2               ************************************************          $
2BEGIN     FOR C6 = (1,1,RA1)                                             $
2COMMENT   4    ************************************************
2          IF THE FOLLOWING TEST IS PASSED, THEN IT HAS BEEN DETERMINED
2          THAT ONE ROW (C1) HAS PREVIOUSLY COMBINED WITH ANOTHER ROW
2          (C6). IT IS NECESSARY TO TEST ROW C2 TO DETERMINE IF IT HAS
2          PREVIOUSLY BEEN MERGED OR COMBINED WITH C6.
2               ************************************************          $
2BEGIN     EITHER IF A2(C1,C6) EQL 1                                      $
2BEGIN     EITHER IF A2(C2,C6) EQL 3                                      $
2BEGIN     IF (C6 EQL C1) AND (C6 EQL C2)                                 $
2          GO PP120                     END                               $
2          OR IF A2(C2,C6) EQL 1                                          $
2BEGIN     FOR C3 = (1,1,RA1)                                             $
2BEGIN     IF A2(C2,C3) EQL 1                                             $
2BEGIN     IF ((C6 NEQ C1) AND (C3 NEQ C2)) OR
2                         ((C6 NEQ L3) AND (C3 NEQ L4))                   $
2BEGIN     T5 = C6                                                        $
2          T6 = C3                                                        $
2          ENTER CHCKMERG                                                 $
2          IF NO                                                          $
2          GO PP100                                                       $
2          ENTER TABLE21                                                  $
2                          END        END        END        END           $
2COMMENT   5    ************************************************
2          IT HAS NOW BEEN DETERMINED THAT C1 HAS BEEN COMBINED WITH
2          C6, AND THAT C2 HAS NOT BEEN MERGED NOR COMBINED WITH C6.
2          EACH  ROW WHICH HAS BEEN COMBINED WITH C2 IS NOW TESTED TO
2          ASSURE THAT IT FULFILLS REQUIREMENT 1, AS LISTED ABOVE.
2               ************************************************          $
```

```
2          OTHERWISE                                                          $
2BEGIN     FOR C3 = (1,1,RA1)                                                 $
2BEGIN     IF A2(C2,C3) EQL 1                                                 $
2BEGIN     IF (C2 NEQ C3) OR (C1 NEQ C6)                                      $
2BEGIN     T5 = C6                                                            $
2          T6 = C3                                                            $
2          ENTER CHCKMERG                                                     $
2          IF NO                                                              $
2          GO PP100                                                           $
2          ENTER TABLE21                                                      $
2COMMENT 6     ***************************************************
2          THIS CONCLUDES PART1 OF THE SUBROUTINE. THE ABOVE TESTS ARE
2          REPEATED FOR ALL ROWS WHICH HAVE BEEN COMBINED WITH ROW C1.
2              **************************************************     $
2PP120..             END        END        END        END        END      $
2COMMENT 7     ***************************************************
2          PART2 OF THE SUBROUTINE BEGINS HERE,  IT IS CONCERNED WITH
2          TESTING RULES 2 AND 3 AS STATED PREVIOUSLY.
2              **************************************************     $
2          OR IF (A2(C1,C6) EQL 3) AND COMB                                   $
2BEGIN     EITHER IF A2(C2,C6) EQL 3                                          $
2          GO PP121                                                           $
2          OR IF A2(C2,C6) EQL 0                                              $
2BEGIN     EITHER IF A6(C6) GTR 0                                             $
2          C8 = A6(C6)                                                        $
2          OTHERWISE                                                          $
2          C8 = C6                                                            $
2          FOR C3 = (1,1,RA1)                                                 $
2BEGIN     IF A2(C8,C3) EQL 1                                                 $
2BEGIN     FOR C7 = (1,1,RA1)                                                 $
2BEGIN     IF A2(C2,C7) EQL 1                                                 $
2BEGIN     T5 = C3                                                            $
2          T6 = C7                                                            $
2          ENTER CHCKMERG                                                     $
```

```
2         IF NO                                                           $
2         GO PP100                                                        $
2         ENTER TABLE21                                                   $
2PP121..                     END        END        END        END         $
2                  END       END        END        END        END         $
2PP100.. RETURN END CHCKCOMB                                              $
2SUBROUTINE TABLE21                                                       $
2COMMENT 8     *****************************************************
22        DURING ANY TEST OF A POSSIBLE MERGER, IT IS NECESSARY TO CHECK
2         ALL RESTRICTIONS TO SEE IF THEY MAY BE SATISFIED. ALL SUCH
2         RESTRICTIONS DURING ANY TRIAL ARE LISTED IN MATRIX A4(), AND
2         THE NUMBER OF LISTINGS FOR ANY TRIAL IS RECORDED AS THE
2         VARIABLE RA4. THE FOLLOWING SUBROUTINE WILL LIST ANY
2         RESTRICTION, FOR A TRIAL, IN MATRIX A4(), AFTER FIRST CHECKING
2         TO ASSURE THAT  THE RESTRICTION HAS NOT BEEN PREVIOUSLY
2         LISTED.ALL RESTRICTIONS ARE LISTED  WITH THE SAME FORMAT AS
2         WAS USED TO LIST THEM IN MATRIX A9(,),(COMMENT 9 OF PART1).
2         THE NECESSARY INPUT TO THIS SUBROUTINE IS THE IDENTIFIER T1,
2         WHICH SPECIFIES THE ROW NUMBER OF MATRIX A9(,) WHICH CONTAINS
2         THE POSSIBLE MERGER BEING CHECKED FOR RESTRICTIONS.
2              ****************************************************       $
2BEGIN    T2 = A9(T1,2) + 2                                               $
2         FOR T3 = (3,1,T2)                                               $
2BEGIN    T4 = A9(T1,T3)                                                  $
2         FOR T5 = (1,1,RA4)                                              $
2BEGIN    IF T4 EQL A4(T5)                                                $
2         GO TP1                        END                               $
2         RA4 = RA4 + 1                                                   $
2         A4(RA4) = T4                                                    $
2TP1..                                  END                               $
2         RETURN END TABLE21                                              $
2COMMENT 9     *****************************************************
2         THE BUILDUP OF THE MODIFIED MERGER DIAGRAM IS CONTROLLED BY THE
2         FOLLOWING SECTION OF THE PROGRAM. THE TRIAL  OF THE POSSIBLE
```

```
2          MERGER  IS CONDUCTED IN A TWO-PASS OPERATION. THE FIRST PASS IS
2          CONCERNED ONLY WITH POSSIBLE MERGERS WHICH HAVE RESTRICTIONS,
2          WHILE THE SECOND PASS WILL CONSIDER ALL POSSIBLE MERGERS. THE
2          MODIFIED MERGER DIAGRAM WILL LIST ROWS WHICH MAY BE MERGED,
2          AND ROWS WHICH MUST BE COMBINED. THIS INFORMATION IS STORED
2          IN MATRIX A2(,) AS ZEROS, ONES AND THREES. FOR THIS OPERATION,
2          MATRIX A2(,) WILL BE A SQUARE TWO DEMSIONAL MATRIX WITH THE
2          NUMBER OF ROWS AND COLUMNS REQUIRED BEING EQUAL TO THE NUMBER
2          OF ROWS IN THE ORIGONAL  FLOW MATRIX. THE INTERSECTION OF ROW
2          I AND COLUMN J OF MATRIX A2(,) INDICATES THE CONDITIONS
2          IMPOSED UPON ROW I AND ROW J OF THE FLOW MATRIX WITH THE
2          FOLLOWING CODE,  0 - ROWS WILL NOT MERGE, 1 - ROWS MUST BE
2          COMBINED, 3 - ROWS MAY BE MERGED IN THE REDUCED FLOW
2          MATRIX.  IT IS SEEN THAT MATRIX A2(,) WILL BE SYMMETRICAL
2          ABOUT ITS DIAGONAL.
2          DURING A TRIAL OF A POSSIBLE MERGER, MATRIX A5(,) WILL BE USED
2          TO STORE TEMPORARY INFORMATION CONCERNING THE MODIFIED MERGER
2          DIAGRAM, WHILE MATRIX A2(,) WILL STORE ONLY THE RESULTS OF ALL
2          SUCCESSFUL TRIALS. AT THE BEGINING OF A TRIAL, MATRIX A5(,)
2          AND A2(,) WILL CONTAIN THE SAME INFORMATION (IN THE SAME
2          FORMAT). A TRIAL WILL BE CONDUCTED ON A POSSIBLE MERGER AND
2          ALL ASSOCIATED RESTRICTIONS. THE POSSIBLE MERGER IS FIRST
2          TESTED, AND IF     IT IS ACCEPTED, IT IS THEN LISTED AS
2          SUCCESSFUL IN MATRIX A5(,). ALL ASSOCIATED RESTRICTIONS ARE
2          THEN CHECKED TO DETERMINE IF THE PAIR OF ROWS WHICH FORM THE
2          RESTRICTION MAY BE COMBINED. IF A RESTRICTION IS FOUND  TO
2          BE COMBINABLE, IT IS SO INDICATED IN MATRIX A5(,).
2          SHOULD ALL RESTRICTIONS BE SATISFIED, THEN THE TRIAL IS
2          SUCCESSFUL AND MATRIX A2(,) IS CHANGED TO AGREE WITH MATRIX
2          A5(,). IF HOWEVER, ANY RESTRICTION IS FOUND WHICH IS IMPOSSIBLE
2          TO SATISFY,   THEN MATRIX A5(,) IS CORRECTED FROM MATRIX A2(,),
2          AND A NEW TRIAL IS CONSIDERED.
2               *****************************************************     $
2PP125.. FOR I = (1,1,RA1)                                                 $
```

```
2BEGIN   FOR J = (1,1,RA1)                                                   $
2BEGIN   A5(I,J) = A2(I,J) = 0          END        END                       $
2COMMENT 10     *****************************************************
2        MATRICES A5(,) AND A2(,) HAVE NOW BEEN CLEARED TO ZERO, AND THE
2        DIAGONAL OF EACH MATRIX WILL BE SET EQUAL TO 1. THIS WILL
2        INDICATE THAT ROW I WILL MERGE WITH ROW I.    MATRICES A4(),
2        A6() AND A7() ARE ALSO CLEARED TO ZERO.
2               *****************************************************        $
2        FOR I = (1,1,RA1)                                                   $
2BEGIN   A2(I,I) = A5(I,I) = 1                                               $
2        A4(I) = A6(I) = A7(I) = 0                 END                       $
2COMMENT 11     *****************************************************
2        THE FIRST PASS THROUGH MATRIX A9(,) WILL NOW BE MADE. A BOOLEAN
2        CODE NO1 IS SET EQUAL TO 0 TO INDICATE THAT ONLY POSSIBLE
2        MERGERS WITH RESTRICTIONS WILL BE CONSIDERED. AT THE COMPLETION
2        OF THE FIRST PASS, NO1 WILL BE CHANGED TO 1 AND THE FOLLOWING
2        SEGMENT OF THE PROGRAM REPEATED.
2               *****************************************************        $
2        NO1 = 0                                                             $
2COMMENT 12     *****************************************************
2        EACH NEW TRIAL BEGINS AT THIS POINT WITH THE SELECTION OF A
2        POSSIBLE MERGER FROM MATRIX A9(,). THE SELECTION IS MADE IN
2        THE ORDER OF LISTING, ACCORDING TO THE CODE NO1.
2               *****************************************************        $
2PP141.. FOR I = (1,1,RA9)                                                   $
2BEGIN   IF (A9(I,2) GTR 0) OR NO1                                           $
2BEGIN   RA4 = 0                                                             $
2COMMENT 13     *****************************************************
2        NO IS A BOOLEAN CODE WHICH WILL BE SET EQUAL TO ZERO BEFORE
2        AN INDIVIDUAL RESTRICTION IS CONSIDERED. IF AT SOME  POINT
2        DURING THE TRIAL IT IS FOUND THAT ALL RESTRICTIONS MAY NOT BE
2        SATISFIED, THEN NO IS SET EQUAL TO 1.
2               *****************************************************        $
2        NO = 0                                                              $
```

```
2COMMENT 14    ************************************************************
2         THE POSSIBLE MERGER WHICH IS BEING TESTED DURING THIS TRIAL
2         WILL BE SELECTED FROM MATRIX A9(,) AND SEPARATED INTO THE
2         INDIVIDUAL ROW NUMBERS(L3 AND L4). THESE ARE THE ROW NUMBERS OF
2         THE FLOW MATRIX WHICH ARE BEING CONSIDERED AS POSSIBLE MERGERS
2              *******************************************************        $
2         L = A9(I,1)                                                         $
2         T1 = I                                                              $
2         L3 = L/1000                                                         $
2         L4 = MOD(L,1000)                                                    $
2COMMENT 15    ************************************************************
2         WHENEVER A ROW HAS BEEN COMBINED WITH ANOTHER ROW, THE
2         COMBINATION IS RECORDED IN MATRIX A6() IN THE FOLLOWING WAY.
2         IF ROW I MUST COMBINE WITH ROW J. THEN THE NUMBER J IS RECORDED
2         IN ROW I OF MATRIX A6(),(THAT IS A6(I) = J). IN ALL CASES OF
2         COMBINATIONS, THE LARGER ROW NUMBERS ARE LISTED AS COMBINING
2         WITH THE LOWEST ROW NUMBERS IF AT SOME LATER TRIAL, ROW J IS
2         COMBINED WITH ROW L, THEN MATRIX A6() IS CHANGED TO READ
2         A6(J) = L AND A6(I) = L, WHERE L IS A ROW NUMBER LESS THAN ROW
2         NUMBER J. HOWEVER, IF THE NUMBER J IS LESS THAN THE
2         NUMBER A6(L),      THEN THE PROGRAM SETS A6(L) = J.
2         A TEST IS NOW MADE TO DETERMINE IF EITHER OR BOTH OF THE ROWS
2         TO BE TESTED (L3 AND L4) HAVE BEEN COMBINED WITH OTHER ROWS. IF
2         A ROW HAS PREVIOUSLY BEEN COMBINED WITH ANOTHER ROW, IT IS
2         NECESSARY TO TEST THE POSSIBLE MERGER USING THE NUMBER OF THE
2         ROW WITH WHICH IT HAS COMBINED RATHER  THAN ITS OWN NUMBER.
2         IN ANY CASE,THE ROWS TO BE TESTED ARE NOW DESIGNATED AS  L1 AND
2         L2.
2              *******************************************************        $
2         EITHER IF A6(L3) GTR 0                                              $
2         L1 = A6(L3)                                                         $
2         OTHERWISE                                                           $
2         L1 = L3                                                             $
2         EITHER IF A6(L4) GTR 0                                              $
```

```
2          L2 = A6(L3)                                                  $
2          OTHERWISE                                                    $
2          L2 = L4                                                      $
2COMMENT 16     ***********************************************************
2          A TEST IS MADE TO DETERMINE IF THE ROWS CONSIDERED AS POSSIBLE
2          MERGERS HAVE EITHER BEEN INDICATED AS COMBINED OR MERGED WITH
2          EACH OTHER. IF EITHER ACTION HAS BEEN TAKEN, IT IS THEN NO
2          LONGER NECESSARY TO CONTINUE THE TRIAL, AND IT IS TERMINATED.
2               *******************************************************   $
2          IF A2(L1,L2) GTR 0                                           $
2          GO PP180                                                     $
2COMMENT 17     ***********************************************************
2          IT HAS BEEN DETERMINED THAT THE TRIAL WILL CONTINUE, THEREFORE
2          SUBROUTINE TABLE21 IS NOW ENTERED AND ALL RESTRICTIONS
2          ASSOCIATED DIRECTLY WITH THE MERGER BEING CONSIDERED ARE
2          RECORDED IN MATRIX A4().
2               *******************************************************   $
2          T1 = 1                                                       $
2          ENTER TABLE21                                                $
2COMMENT 18     ***********************************************************
2          IF EITHER OR BOTH OF THE ROWS TO BE MERGED HAS COMBINED
2          WITH ONE OR MORE OTHER ROWS, A CHECK MUST BE CONDUCTED TO
2          DETERMINE IF THE TWO ROWS NOW BEING CONSIDERED AS POSSIBLE
2          MERGERS ARE STILL COMPATIBLE.
2          INCOMPATIBILITY WOULD RESULT IN CASE ROW I HAS PREVIOUSLY
2          COMBINED WITH ROW L, AND ROW J WHICH COULD MERGE WITH ROW I
2          WILL NOT MERGE WITH ROW L. THIS CHECK IS PERFORMED IN
2          SUBROUTINE CHCKMERG.
2               *******************************************************   $
2          IF L2 LSS L1                                                 $
2BEGIN     T1 = L1                                                      $
2          T2 = L2                                                      $
2          L1 = T2                                                      $
2          L2 = T1                     END                              $
```

```
2         T5 = L1                                                      $
2         T6 = L2                                                      $
2         ENTER CHCKMERG                                               $
2COMMENT  19    *****************************************************
2         TWO COURSES OF ACTION ARE NOW POSSIBLE. IF THE ROWS HAVE BEEN
2         FOUND TO BE INCOMPATIBLE,THEN MATRIX A5(,) MUST BE RESET FOR A
2         A NEW TRIAL. OTHERWISE THE RESTRICTIONS UPON THE NEW
2         COMBINATION OF ROWS MUST BE LISTED IN MATRIX A4().
2               ****************************************************   $
2         IF NO                                                        $
2         GO PP180                                                     $
2         ENTER TABLE21                                                $
2COMMENT  20    *****************************************************
2         THE TRIAL WILL CONTINUE, AND  SUBROUTINE CHCKCOMB IS NOW
2         ENTERED. COMB IS A BOOLEAN CODE SET EQUAL TO ZERO TO INDICATE
2         THAT THE TWO ROWS BEING TESTED ARE TO BE MERGED RATHER  THAN
2         COMBINED. A CHANGE OF VARIABLES IS NECESSARY FOR THE
2         SUBROUTINE.
2               ****************************************************   $
2         C1 = L1                                                      $
2         C2 = L2                                                      $
2         COMB = 0                                                     $
2         ENTER CHCKCOMB                                               $
2COMMENT  21    *****************************************************
2         TWO CHOICES ARE NOW AVAILABLE TO THE PROGRAM. IF THE SUBROUTINE
2         DETERMINED THAT THE TWO ROWS MAY NOT MERGE, THEN THE TRIAL IS
2         TERMINATED (NO = 1). OTHERWISE THE TWO ROWS ARE LISTED AS
2         POSSIBLE MERGERS IN THE MODIFIED MERGER DIAGRAM. THIS IS
2         SHOWN BY CHANGING THE INTERSECTION OF THE TWO ROWS IN MATRIX
2         A5(,) FROM 0 TO 3.
2               ****************************************************   $
2         IF NO                                                        $
2         GO PP199                                                     $
2         A2(L1,L2) = A2(L2,L1) = 3                                    $
```

```
2COMMENT 22    *******************************************************
2        AT THIS POINT, THE TWO ROWS UNDER CONSIDERATION AS POSSIBLE
2        MERGERS HAVE BEEN CHECKED AND ACCEPTED. IT IS NOW  NECESSARY
2        TO EXAMINE ALL REQUIREMENTS ASSOCIATED WITH THE TWO ROWS AND
2        DETERMINE IF THEY MAY BE SATISFIED.  ALL RESTRICTIONS WHICH
2        HAVE BEEN DETERMINED SO FAR FOR THIS TRIAL ARE LISTED IN
2        MATRIX A4().  EACH WILL NOW BE EXAMINED TO DETERMINE IF
2        THEY MAY BE COMBINED. AS EACH RESTRICTION IS ACCEPTED, IT IS
2        RECORDED AS BEING COMBINED IN MATRIX A5(,) AND MATRIX A4().
2        ANY NEW RESTRICTION(S) UPON THE RESTRICTION BEING TESTED IS
2        ADDED TO MATRIX A4().
2        SHOULD ANY RESTRICTION, LISTED IN MATRIX A4(), FAIL
2        ANY OF THE TESTS, THEN THIS TRIAL IS TERMINATED AS A FAILURE.
2              *****************************************************     $
2        FOR I1 = (1,1,RA4)                                              $
2BEGIN   K = A4(I1)                                                      $
2COMMENT 23    *******************************************************
2        AS THE RESTRICTIONS ARE DRAWN FROM MATRIX A4(), THEY ARE
2        FIRST TESTED TO DETERMINE IF ANY RESTRICTIONS EXIST UPON
2        THEMSELVES WHEN LISTED AS A POSSIBLE MERGER. IF ANY
2        RESTRICTIONS ARE FOUND, THEY ARE ADDED TO MATRIX A4() BY
2        SUBROUTINE TABLE21.
2              *****************************************************     $
2        FOR I2 = (1,1,RA9)                                              $
2BEGIN   IF K EQL A9(I2,1)                                               $
2BEGIN   T1 = I2                                                         $
2        ENTER TABLE21                                                   $
2        GO PP140                        END       END                   $
2COMMENT 24    *******************************************************
2        THE RESTRICTION BEING TESTED IS NOW BROKEN  INTO ITS TWO
2        PARTS AND TESTED AGAINST MATRIX A6() IN THE SAME WAY AS
2        INDICATED IN COMMENT 15.
2              *****************************************************     $
2PP140.. L3 = K/1000                                                     $
```

```
2          L4 = MOD(K,1000)                                              $
2          EITHER IF A6(L3) GTR 0                                        $
2          K1 = A6(L3)                                                   $
2          OTHERWISE                                                     $
2          K1 = L3                                                       $
2          EITHER IF A6(L4) GTR 0                                        $
2          K2 = A6(L4)                                                   $
2          OTHERWISE                                                     $
2          K2 = L4                                                       $
2COMMENT   25   ****************************************************
2          A TEST IS MADE TO DETERMINE IF THE TWO ROWS BEING TESTED HAVE
2          ALREADY BEEN INDICATED  AS REQUIRED COMBINATIONS. IF THEY
2          HAVE BEEN SO INDICATED, THEN NO TESTS ARE NECESSARY UPON THIS
2          RESTRICTION.
2               ****************************************************     $
2          IF A2(K1,K2) EQL 1                                            $
2          GO PP181                                                      $
2COMMENT   26   ****************************************************
2          A TEST IS MADE TO DETERMINE IF THE TWO ROWS UNDER CONSIDERATION
2          HAVE BEEN LISTED AS POSSIBLE MERGERS IN THE MERGER
2          DIAGRAM. IF THIS IS TRUE,THEN THE PROCESS IS TERMINATED
2          AS A FAILURE. THIS TRIAL IS THEN CONCLUDED TO BE A FAILURE.
2          IT IS NOT NECESSARY TO CONCLUDE THE TRIAL AT THIS POINT SINCE
2          TWO ROWS WHICH HAVE BEEN INDICATED AS POSSIBLE MERGERS COULD
2          BE TESTED TO DETERMINE IF THEY MAY BE COMBINED WITHOUT
2          CHANGING POSSIBLE MERGERS CONNECTED TO EACH ROW.  HOWEVER
2          SINCE THE ADDITIONAL TESTING AND COMBINING OF THE ROWS WOULD
2          LENGTHEN THIS SEGMENT OF THE PROGRAM CONSIDERABLY, AND DUE TO
2          THE LACK OF STORAGE SPACE, NO SUCH PROGRAM HAS BEEN ADDED.
2               ****************************************************     $
2          IF A2(K1,K2) EQL 3                                            $
2BEGIN     NO = 1                                                        $
2          GO PP199                         END                          $
2COMMENT   27   ****************************************************
```

```
2          A TEST IS MADE TO DETERMINE IF THE HIGHEST NUMBER ROW TO BE
2          COMBINED IS THE SAME AS THE HIGHEST NUMBER ROW WHICH FORMS THE
2          POSSIBLE MERGER. IF THIS IS TRUE, THEN THE TRIAL WILL BE
2          TERMINATED AS A FAILURE. IF THE TRIAL IS POSSIBLE, THEN THE
2          MERGERS AND COMBINATIONS WILL BE DETERMINED AT A LATER TRIAL
2          WHEN ROW L1 (THE LOWEST NUMBER ROW TO BE MERGED) IS TESTED
2          AGAINST ROW L3 (THE LOWEST NUMBER ROW TO BE COMBINED).
2                *****************************************************     $
2          IF K2 LSS K1                                                     $
2BEGIN     T1 = K1                                                          $
2          T2 = K2                                                          $
2          K1 = T2                                                          $
2          K2 = T1                         END                              $
2          IF L4 EQL L2                                                     $
2BEGIN     NO = 1                                                           $
2          GO PP199                        END                              $
2          T5 = K1                                                          $
2          T6 = K2                                                          $
2          ENTER CHCKMERG                                                   $
2          IF NO                                                            $
2          GO PP199                                                         $
2COMMENT   28    ******************************************************
2          COMB IS SET EQUAL TO 1 TO INDICATE THAT THE TWO ROWS BEING
2          CONSIDERED ARE TO BE COMBINED. A CHANGE OF VARIABLES IS MADE,
2          AND SUBROUTINE CHCKCOMB IS ENTERED.
2                *****************************************************     $
2          COMB = 1                                                         $
2          C1 = K1                                                          $
2          C2 = K2                                                          $
2          ENTER CHCKCOMB                                                   $
2COMMENT   29    ******************************************************
2          IF THE SUBROUTINE FOUND THAT THE ROWS MAY NOT BE COMBINED, THEN
2          THE TRIAL IS TERMINATED.
2                *****************************************************     $
```

```
2          IF NO                                                              $
2          GO PP199                                                           $
2COMMENT   30    ****************************************************
2          SINCE THE TWO ROWS WERE FOUND TO BE COMBINABLE, THE PROPER
2          CHANGES ARE MADE IN MATRIX A6() ACCORDING TO COMMENT 15, AND
2          THE COMBINATION IS INDICATED IN MATRIX A5(,).
2          NOTE THAT THE HIGHEST NUMBER ROWS ARE ALWAYS SHOWN AS COMBINING
2          WITH THE LOWEST NUMBER ROWS IN MATRIX A6(), ACCORDING TO
2          COMMENT 15.
2                ***************************************************          $
2          A6(K2)= K1                                                         $
2          FOR I3 = (1,1,RA1)                                                 $
2BEGIN     IF A6(I3) EQL K2                                                   $
2          A6(I3) = K1                     END                                $
2          A2(K2,K1) = A2(K1,K2) = 1                                          $
2          IF L1 EQL L4                                                       $
2BEGIN     A2(L1,L2) = A2(L2,L1) = 0                                          $
2          A2(L2,L3) = A2(L3,L2) = 3       END                                $
2COMMENT   31    ****************************************************
2          A RESTRICTION HAS BEEN SUCCESSFULLY CHECKED AS INDICATED IN
2          MATRICES A5(,) AND A6(). THE PROGRAM WILL NOW TEST THE NEXT
2          RESTRICTION LISTED IN MATRIX A4() OR, IF ALL RESTRICTIONS HAVE
2          BEEN CHECKED THE PROGRAM WILL ADVANCE TO THE NEXT SECTION.
2                ***************************************************          $
2PP181..                                   END                                $
2COMMENT   32    ****************************************************
2          THE PROGRAM ARRIVES AT THIS POINT THROUGH EITHER OF TWO
2          OPERATIONS.  EITHER THE TRIAL HAS BEEN COMPLETED AND THE MERGER
2          AND ALL RESTRICTIONS HAVEBEEN INDICATED IN THE PROPER MATRICES,
2          OR THE TRIAL HAS FAILED ONE TEST ( NO WAS SET EQUAL TO 1, AND
2          THE PROGRAM JUMPED TO THIS POINT). IN EITHER CASE, IT IS NOW
2          NECESSARY TO CORRECT THE PROPER MATRICES.IF NO EQUALS ONE, THEN
2          MATRIX A5(,) AND MATRIX A6() ARE CORRECTED ACCORDING TO
2          MATRICES A2(,) AND A7() RESPECTIVELY. IF THE TRIAL WAS
```

```
2          SUCCESSFUL (NO = 0), THEN MATRICES A2(,) AND A7() ARE MODIFIED
2          ACCORDING TO MATRICES A5(,) AND A6(), RESPECTIVELY.
2                ****************************************************       $
2PP199..   EITHER IF NO                                                     $
2BEGIN     FOR I1 = (1,1,RA1)                                               $
2BEGIN     A6(I1) = A7(I1)                                                  $
2          FOR I2 = (1,1,RA1)                                               $
2          A2(I1,I2) = A5(I1,I2)          END        END                    $
2          OTHERWISE                                                        $
2BEGIN     FOR I1 = (1,1,RA1)                                               $
2BEGIN     A7(I1) = A6(I1)                                                  $
2          FOR I2 = (1,1,RA1)                                               $
2          A5(I1,I2) = A2(I1,I2)          END        END                    $
2PP180..                                  END        END                    $
2COMMENT   33    ****************************************************
2          AT THIS POINT, ALL TRIALS OF THE TYPE BEING CONSIDERED HAVE
2          BEEN TESTED.A TEST IS NOW MADE TO DETERMINE WHICH CLASS HAS
2          BEEN TESTED. IF NO1 IS NOW EQUAL TO ZERO, IT IS CHANGED TO ONE
2          TO INDICATE THAT ALL MERGERS ARE TO BE TESTED. THE PROGRAM
2          THEN JUMPS TO COMMENT 12 AND THE TRIAL  PROCESS IS REPEATED.
2          OTHERWISE, ALL THE TRIALS ARE COMPLETED.
2                ****************************************************       $
2          IF NO1 EQL 0                                                     $
2BEGIN     NO1 = 1                                                          $
2          GO PP141                       END                               $
2COMMENT   34    ****************************************************
2          THE MODIFIED MERGER DIAGRAM HAS BEEN DETERMINED AT THIS
2          POINT. IT IS NECESSARY HOWEVER TO CHANGE THE  FORMAT OF THE
2          DIAGRAM TO A FORMAT ACCEPTABLE BY THE NEXT SEGMENT. THE
2          MODIFIED MERGER DIAGRAM IS NOW CONTAINED IN MATRIX A5(,)
2          AND MATRIX A7().    IT WILL BE REWRITTEN IN MATRIX A2(,).
2          THE NEXT COMMANDS SET MATRIX A2(,) EQUAL TO ZERO.
2                ****************************************************       $
2          FOR I = (1,1,RA1)                                                $
```

```
2BEGIN   FOR J = (1,1,RA1)                                                   $
2        A2(I,J) = 0                        END                              $
2COMMENT 35    **********************************************************
2        THE FORMAT OF THE MODIFIED MERGER DIAGRAM IN MATRIX A2(,)
2        WILL BE AS FOLLOWS.  THE ROW NUMBER OF MATRIX A2(,) WILL
2        CORRESPOND TO THE ROW NUMBER OF THE FLOW MATRIX.  THE FIRST
2        COLUMN OF MATRIX A2(,) WILL BE AN INTEGER WHICH INDICATES
2        THE NUMBER OF ROWS WHICH MAY MERGE WITH THE ROW ACCORDING TO
2        THE RESULTS OF THE MATRIX A5(,).  THE NEXT COLUMNS WILL
2        CONTAIN THE ROW NUMBERS OF THE POSSIBLE MERGERS.  IN CASE
2        A ROW HAS BEEN INDICATED AS CONBINING WITH ANOTHER ROW,THEN THE
2        FIRST COLUMN OF MATRIX A2(,) WILL CONTAIN THE THE ROW NUMBER,
2        WITH WHICH IT MUST COMBINE, MULTIPLED BY 1000.
2              *********************************************************   $
2        FOR I = (1,1,RA1)                                                   $
2BEGIN   EITHER IF A7(I) EQL 0                                               $
2BEGIN   FOR I1 = (1,1,RA1)                                                  $
2BEGIN   IF A5(I,I1) EQL 3                                                   $
2BEGIN   A2(I,1) = A2(I,1) + 1                                               $
2        K = A2(I,1) + 1                                                     $
2        A2(I,K) = I1                       END       END       END          $
2        OTHERWISE                                                           $
2        A2(I,1) = 1000.A7(I)               END                              $
2COMMENT 36    **********************************************************
2        THE MODIFIED MERGER DIAGRAM IS NOW PRINTED ON THE LINE PRINTER
2        ACCORDING TO THE FOLLOWING FORMATS AND TITLES.
2              *********************************************************   $
2        WRITE($$TITL2)                                                      $
2FORMAT  TITL2(*CONNECTION MATRIX*,W4,
2              *ROW  NO. OF MERGERS   NUMBER OF THE ROW(S) FOR*,W0,
2              *NO.      POSSIBLE     WHICH A MERGER IS POSSIBLE*,W0)      $
2        MATRIXPRINT1(RA1,RA1,A2(,),2)                                       $
2COMMENT 37    **********************************************************
2        THIS CONCLUDES PART3. THE REQUIRED OUTPUT OF THIS SEGMENT IS
```

```
2          MATRIX A2(,). THE FORMAT IS AS EXPLAINED IN COMMENT 35.
2          THE PROGRAM CONTROL IS NOW GIVEN BACK TO THE MASTER PROGRAM.
2               ************************************************            $
2          GO PP71                                                          $
2          END PART3                                                        $
```

Segment Four: Determination of the Mergers

The purpose of Segment Part 4 is to determine the optimum mergers, given the merger diagram established in Part 3. In order to do this automatically, it was necessary to develop several rules of merging. The basic idea behind these rules is that certain mergers may be performed, and their presence removed from the merger diagram, without destroying the optimum merger property of the given merger diagram. That is, after an optimum merger is removed from the merger diagram, the minimum number of states in a reduced flow matrix may still be determined if the remaining states are also optimally merged. The optimum merger rules used within this segment are now explained, and the program is shown in Figure 32.

Rule One

Figure 28 shows a partial merger which illustrates the application of Rule One. In the given example, state A may merge with only one other state, while state B will merge with either state A or state C. For this example, states A and B may be merged without affecting the optimum merger property of the remaining merger diagram. Similarly states C and D may be removed and merged. It is necessary, however, that the states always be removed in pairs, beginning at the last state (in this case, state A).

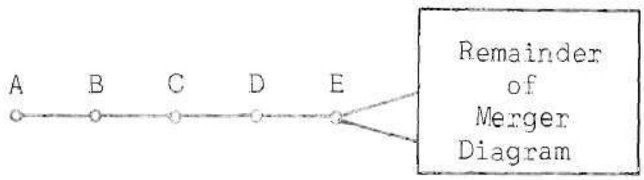

Figure 28. Partial Merger Diagram Illustrating the Application of Merger Rule One

Rule Two

Rule Two considers two situations which may occur within the merger diagram as illustrated in Figures 29(a) and 29(b). The illustration of Figure 29(a) is a special case of the illustration of Figure 29(b).

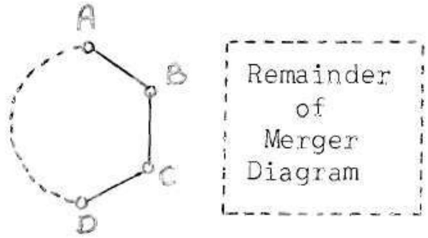

Figure 29(a). Partial Merger Diagram Illustrating the Application of Merger Rule Two

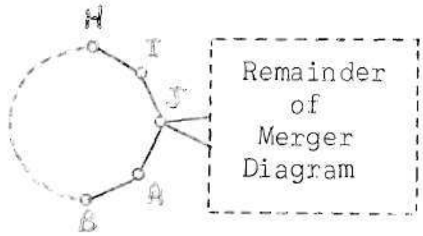

Figure 29(b). Partial Merger Diagram Illustrating the Application of Merger Rule Two

The basic situation is that four or more states are connected together so that no more than one state is connected to the remainder of the merger diagram [as illustrated by state J of Figure 29(b)], and each other state connects to only two other states.

For the independent case of Figure 29(a), it is possible to merge as many pairs of states as exist within the chain. If the chain contains an odd number of states, then one state of the chain will not merge with any other state within the reduced merger diagram; however, all states of the chain will be removed from the merger diagram.

A special case of Figure 29(a) exists when the chain contains only three states. Then it is possible to merge all three states as a single state. This situation is covered by Rule Four.

For the example of Figure 29(b), it is necessary to remove only pairs of states, beginning with a state which is adjacent to state J.

If the number of states contained within the particle chain A through I is even, then all pairs of states will be removed; however, if the number of states is odd, then only the pairs of states A through H will be removed, and State I is left in the merger diagram and its connection to state J is still indicated in the merger diagram.

Rule Three

The application of Rule Three will remove single states from the merger diagram which do not affect the rest of the diagram. An illustration of such a single state is shown in Figure 30.

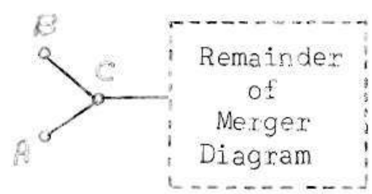

Figure 30. Partial Merger Diagram Illustrating the Application of Merger Rule Three

In Figure 30, state A is a single state connected to only one other state (state C) of the complete merger diagram. State A may be removed from the merger diagram if there exists a single state B which also connects only to state C.

Rule Four

Rule Four may be applied to the type of merger illustrated by Figure 31. There, state A merges with two or more states of the remainder of the merger diagram. The states to which A may be merged are known and are recorded. It is then necessary to examine each state of the list to determine whether it will merge with all other states of the listing. If this is possible, then state A along with all states to which it connects may be merged as a single state in the reduced flow matrix.

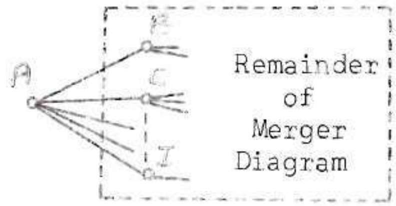

Figure 31. Partial Merger Diagram Illustrating the Application of Merger Rule Four

The states which are merged by this rule are then removed from the merger diagram and the remainder of the diagram examined for other reductions.

Trial Examination Within Segment Four

If repeated application of the four rules previously outlined fail to completely reduce a merger diagram, then it is concluded that a situation has been found which is not covered by the rules listed so far.

It is now necessary to either establish new optimum merger rules, or to initiate a trial reduction procedure. It was found, however, that these four rules successively reduced all the examples given in Appendix B.

The program for Segment Part 4 is shown in Figure 32, and the required output of this program is explained in comment 12. A printout of the output of the segment is illustrated in Appendix B.

```
2SEGMENT PART4                                   $          BEGIN
2COMMENT 1    *****************************************************
2        SEGMENT PART4 BEGINS HERE. THE PURPOSE OF PART4 IS TO DETERMINE
2        THE OPTIMUM MERGER POSSIBLE WITH RESPECT TO THE MERGER DIAGRAM
2        FOUND IN SEGMENT PART3. SEVERAL RULES ARE SET FORTH, ALL OF
2        WHICH RESULT IN OPTIMUM MERGER OF A MERGER DIAGRAM. THE
2        APPLICATION OF THESE RULES IS  EXPLAINED IN THE TEXT.
2        AGAIN, SEVERAL SUBROUTINES ARE USED.
2             ****************************************************      $
2SUBROUTINE COMBINE2                                                    $
2BEGIN   SWITCH C,(ONS,TWS,THS,FOS)                                     $
2ONS..   K = 0                                                          $
2        IF (2.ENTIRE(CA4/2)) LSS CA4                                   $
2BEGIN   K = 1                                                          $
2        RA5 = RA5 + 1                                                  $
2        L = A4(K)                                                      $
2        A5(RA5,1) = L                                                  $
2        A2(L,1) = 0                     END                            $
2PP2..   UNTIL K EQL CA4                                                $
2BEGIN   RA5 = RA5 + 1                                                  $
2        K = K + 1                                                      $
2        L = A4(K)                                                      $
2        A5(RA5,1) = L                                                  $
2        A2(L,1) = 0                                                    $
2        K = K + 1                                                      $
2        L = A4(K)                                                      $
2        A5(RA5,2) = L                                                  $
2        A2(L,1) = 0                     END                            $
2        GO COM                                                         $
2THS..   EITHER IF (2.ENTIRE(CA4/2)) LSS CA4                            $
2BEGIN   R1 = A4(CA4)                                                   $
2        CA4 = CA4 - 1                                                  $
2        R2 = A4(CA4)                    END                            $
2        OTHERWISE                                                      $
```

Figure 32. ALGOL Program of Segment Part 4.
(Continued on the next 11 pages.)

```
2          R2 = A4(CA4)                                                    $
2          ENTER REMOVE                                                    $
2          GO ONS                                                          $
2TWS..                                                                     $
2FOS..                                                                     $
2COM..     RETURN END COMBINE2                                             $
2SUBROUTINE REMOVE                                                         $
2BEGIN     R3 = A2(R1,1) + 1                                               $
2          FOR R4 = (2,1,R3)                                               $
2BEGIN     IF A2(R1,R4) EQL R2                                             $
2          GO RE1                         END                              $
2RE1..     R5 = R4                                                         $
2          UNTIL R5 EQL R3                                                 $
2BEGIN     R5 = R5 + 1                                                     $
2          A2(R1,R4) = A2(R1,R5)                                           $
2          R4 = R4 + 1                    END                              $
2          A2(R1,1) = A2(R1,1) - 1                                         $
2          IF A2(R1,1) EQL 0                                               $
2BEGIN     RA5 = RA5 + 1                                                   $
2          A5(RA5,1) = R1                 END                              $
2          A2(R1,R5) = 0                                                   $
2          RETURN END REMOVE                                               $
2SUBROUTINE COMBINEN                      $
2BEGIN     RA5 = RA5 + 1                                                   $
2          FOR N = (1,1,CA4)                                               $
2BEGIN     L = A4(N)                                                       $
2          A5(RA5,N) = L                                                   $
2          A2(L,1) = 0                    END                              $
2          RETURN END COMBINEN                                             $
2PP72..    FOR I = (1,1,RA1)                                               $
2BEGIN     FOR J = (1,1,RA1)                                               $
2          A5(I,J) = 0                    END                              $
2COMMENT 2    ****************************************************
2          MATRIX A5(,) HAS JUST BEEN CLEARED TO ZERO AND WILL BE USED TO
```

```
2         STORE THE INDIVIDUAL MERGERS AS THEY ARE DETERMINED.EACH ROW
2         OF MATRIX A5(,) WILL STORE A LISTING OF THE ROWS OF THE FLOW
2         MATRIX WHICH FORM A MERGED ROW IN THE REDUCED FLOW MATRIX.
2         AS MANY COLUMNS OF MATRIX A5(,) WILL BE USED AS REQUIRED, WITH
2         A MAXIMUM OF RA1 COLUMNS BEING REQUIRED (WHEN RA1 COLUMNS ARE
2         REQUIRED, THEN THE REDUCED FLOW MATRIX WILL CONSIST OF ONLY
2         ONE ROW). RA5 WILL BE SET EQUAL TO ZERO AND USED TO COUNT THE
2         NUMBER OF ROWS USED IN MATRIX A5(,). THIS WILL BE THE NUMBER OF
2         ROWS IN THE REDUCED FLOW MATRIX. CRT1 IS A COUNTER SET EQUAL TO
2         ZERO. AT THE CONCLUSION OF PART5,A COUNT WILL BE MADE OF THE
2         NUMBER OF MERGERS NOT TRANSFERED TO MATRIX A5(,). IF THE COUNT
2         DOES NOT AGREE WITH CTR1, THEN THE MERGING RULES ARE AGAIN
2         APPLIED AS THE APPLICATION OF CERTAIN RULES ALLOW THE USE OF
2         OTHER RULES WHICH PREVIOUSLY WERE NOT APPLICABLE. THE
2         ELIMINATION PROCESS IS REPEATED AS OFTEN AS IT IS SUCCESSFUL,
2         OR UNTIL THE MERGER DIAGRAM HAS BEEN REDUCED.
2              ***************************************************        $
2         RA5 = 0                                                         $
2         CTR1 = 0                                                        $
2COMMENT 3     ****************************************************
2         THE FIRST REDUCTION OF THE MERGER DIAGRAM IS TO REMOVE ANY
2         ROWS WHICH ARE UNABLE TO MERGE WITH ANOTHER ROW.SUCH ROWS
2         WILL CONTAIN A ZERO IN COLUMN ONE OF MATRIX A2(,). LATER AS
2         THE INDIVIDUAL MERGERS ARE REMOVED FROM THE DIAGRAM, THEIR
2         REMOVAL WILL BE INDICATED BY SETTING THEIR FIRST COLUMN EQUAL
2         TO ZERO. THUS THE REDUCTION PROCESS DESTROYS THE MERGER
2         DIAGRAM.
2              ***************************************************        $
2         FOR I = (1,1,RA1)                                               $
2BEGIN    IF A2(I,1) EQL 0                                                $
2BEGIN    RA5 = RA5 + 1                                                   $
2         A5(RA5,1) = I                     END        END                $
2COMMENT 4     ****************************************************
2         MERGERS WHICH MAY BE REMOVED FROM THE MERGER DIAGRAM ACCORDING
```

```
2           TO RULE 1, ARE NOW REMOVED ACCORDING THE THE FOLLOWING
2           COMMANDS.
2                 ******************************************************      $
2PP6..      FOR I = (1,1,RA1)                                                  $
2BEGIN      IF A2(I,1) EQL 1                                                   $
2BEGIN      CA4 = 0                                                            $
2           B = A2(I,2)                                                        $
2           CA4 = CA4 + 1                                                      $
2           A4(CA4) = I                                                        $
2PP1..      EITHER IF A2(B,1) GTR 2                                            $
2           C = 3                                                              $
2           OR IF A2(B,1) EQL 2                                                $
2BEGIN      C = 2                                                              $
2           L = A4(CA4)                                                        $
2           CA4 = CA4 + 1                                                      $
2           A4(CA4) = B                                                        $
2           B1 = A2(B,2)                                                       $
2           B2 = A2(B,3)                                                       $
2           EITHER IF B1 EQL L                                                 $
2           B = B2                                                             $
2           OTHERWISE                                                          $
2           B = B1                        END                                  $
2           OTHERWISE                                                          $
2BEGIN      C = 1                                                              $
2           CA4 = CA4 + 1                                                      $
2           A4(CA4) = B                   END                                  $
2           EITHER IF C EQL 3                                                  $
2BEGIN      IF CA4 GEQ 2                                                       $
2BEGIN      R1 = B                                                             $
2           ENTER COMBINE2                END          END                     $
2           OR IF C EQL 1                                                      $
2           ENTER COMBINE2                                                     $
2           OTHERWISE                                                          $
2           GO PP1                        END          END                     $
```

```
2COMMENT 5      ******************************************************
2        MERGERS WHICH MAY BE REMOVED FROM THE MERGER DIAGRAM ACCORDING
2        TO RULE 2, ARE NOW REMOVED FROM THE MERGER DIAGRAM ACCORDING
2        TO THE FOLLOWING COMMANDS.
2               *****************************************************    $
2        FOR I = (1,1,RA1)                                               $
2BEGIN   IF A2(I,1) EQL 2                                                $
2BEGIN   CA4 = 1                                                         $
2        A4(CA4) = I                                                     $
2        X1 = A2(I,2)                                                    $
2        X2 = A2(I,3)                                                    $
2PP3..   IF A2(X1,1) EQL 2                                               $
2BEGIN   B1 = A2(X1,2)                                                   $
2        B2 = A2(X1,3)                                                   $
2        EITHER IF B1 EQL A4(CA4)                                        $
2BEGIN   CA4 = CA4 + 1                                                   $
2        A4(CA4) = X1                                                    $
2        X1 = B2                        END                              $
2        OTHERWISE                                                       $
2BEGIN   CA4 = CA4 + 1                                                   $
2        A4(CA4) = X1                                                    $
2        X1 = B1                        END        END                   $
2        EITHER IF A2(X1,1) EQL 1                                        $
2        GO PP6                                                          $
2        OR IF X1 EQL A4(1)                                              $
2        GO PP5                                                          $
2        OR IF A2(X1,1) GTR 2                                            $
2        GO PP4                                                          $
2        OTHERWISE                                                       $
2        GO PP3                                                          $
2PP4..   CA6 = 1                                                         $
2        A6(CA6) = I                                                     $
2PP13..  IF A2(X2,1) EQL 2                                               $
2BEGIN   B1 = A2(X2,2)                                                   $
```

```
2          B2 = A2(X2,3)                                              $
2          EITHER IF B1 EQL A6(CA6)                                   $
2BEGIN     CA6 = CA6 + 1                                              $
2          A6(CA6) = X2                                               $
2          X2 = B2                          END                       $
2          OTHERWISE                                                  $
2BEGIN     CA6 = CA6 + 1                                              $
2          A6(CA6) = X2                                               $
2          X2 = B1                          END                       $
2          IF X2 NEQ A6(1)                                            $
2          GO PP13                          END                       $
2PP5..     EITHER IF X1 EQL A4(1)                                     $
2BEGIN     EITHER IF CA4 GTR 3                                        $
2BEGIN     C = 1                                                      $
2          ENTER COMBINE2                   END                       $
2          OTHERWISE                                                  $
2          ENTER COMBINEN                   END                       $
2          OR IF X1 EQL X2                                            $
2BEGIN     CA7 = CA4 - 1                                              $
2          FOR K = (2,1,CA4)                                          $
2          A7(K) = A4(K)                                              $
2          CA4 = 0                                                    $
2          UNTIL CA6 EQL 0                                            $
2BEGIN     CA4 = CA4 + 1                                              $
2          A4(CA4) = A6(CA6)                                          $
2          CA6 = CA6 - 1                    END                       $
2          FOR K = (1,1,CA7)                                          $
2BEGIN     CA4 = CA4 + 1                                              $
2          A4(CA4) = A7(K)                  END                       $
2          IF CA4 GEQ 3                                               $
2BEGIN     R1 = X2                                                    $
2          R2 = A4(1)                                                 $
2          ENTER REMOVE                                               $
2          EITHER IF CA4 GTR 3                                        $
```

```
2BEGIN   R1 = X1                                              $
2        C = 3                                                $
2        ENTER COMBINE2              END                      $
2        OTHERWISE                                            $
2        ENTER COMBINEN              END       END            $
2        OTHERWISE                                            $
2BEGIN   IF (CA4 + CA6) GTR 2                                 $
2BEGIN   L = A2(X1,1) + 1                                     $
2        FOR K = (2,1,L)                                      $
2BEGIN   IF A2(K,1) EQL 1                                     $
2BEGIN   R1 = X1                                              $
2        R2 = A4(CA4)                                         $
2        ENTER REMOVE                                         $
2        CA7 = CA4 - 1                                        $
2        FOR K = (2,1,CA4)                                    $
2        A7(K-1) = A4(K)                                      $
2        CA4 = 0                                              $
2        UNTIL CA7 EQL 0                                      $
2BEGIN   CA4 = CA4 + 1                                        $
2        A4(CA4) = A7(CA7)                                    $
2        CA7 = CA7 - 1               END                      $
2        FOR K = (1,1,CA6)                                    $
2BEGIN   CA4 = CA4 + 1                                        $
2        A4(CA4) = A6(K)             END                      $
2        C = 3                                                $
2        R1 = X2                                              $
2        ENTER COMBINE2                                       $
2        GO PP12                     END       END            $
2        L = A2(X2,1) + 1                                     $
2        FOR K = (2,1,L)                                      $
2BEGIN   IF A2(K,1) EQL 1                                     $
2BEGIN   R1 = X2                                              $
2        R2 = A6(CA6)                                         $
2        ENTER REMOVE                                         $
```

```
2          CA7 = CA4 - 1                                                   $
2          FOR K = (2,1,CA4)                                               $
2          A7(K-1) = A4(K)                                                 $
2          CA4 = 0                                                         $
2          UNTIL CA6 EQL 0                                                 $
2BEGIN     CA4 = CA4 + 1                                                   $
2          A4(CA4) = A7(CA7)                                               $
2          CA6 = CA6 - 1                END                                $
2          FOR K = (1,1,CA7)                                               $
2BEGIN     CA4 = CA4 + 1                                                   $
2          A4(CA4) = A7(K)              END                                $
2          C = 3                                                           $
2          R1 = X2                                                         $
2          ENTER COMBINE2               END      END                       $
2PP12..                        END      END      END      END              $
2COMMENT 6    *****************************************************
2          MERGERS WHICH MAY BE REMOVED FROM THE MERGER DIAGRAM ACCORDING
2          TO RULE 3, ARE NOW REMOVED ACCORDING TO THE FOLLOWING COMMANDS.
2             ****************************************************         $
2          FOR I = (1,1,RA1)                                               $
2BEGIN     IF (A2(I,1) GTR 2) AND (A2(I,1) LSS 1000)                       $
2BEGIN     L = A2(I,1) + 1                                                 $
2          FOR  J = (2,1,L)                                                $
2BEGIN     K = A2(I,J)                                                     $
2          IF A2(K,1) EQL 1                                                $
2BEGIN     L1 = J + 1                                                      $
2          FOR K1 = (L1,1,L)                                               $
2BEGIN     L2 = A2(I,K1)                                                   $
2          IF A2(L2,1) EQL 1                                               $
2BEGIN     R2 = K                                                          $
2          R1 = I                                                          $
2          ENTER REMOVE                                                    $
2          RA5 = RA5 + 1                                                   $
2          A5(RA5,1) = K1                                                  $
```

```
2           A2(K1,1) = 0
2           END          END          END          END          END          END          $
2COMMENT    7      *****************************************************************
2           MERGERS WHICH MAY BE REMOVED FROM THE MERGER DIAGRAM ACCORDING
2           TO RULE 4, ARE NOW REMOVED ACCORDING TO THE FOLLOWING COMMANDS.
2                  ****************************************************************  $
2           FOR I = (1,1,RA1)                                                        $
2BEGIN      IF (A2(I,1) GEQ 2) AND (A2(I,1) LSS 1000)                                $
2BEGIN      CA4 = 0                                                                  $
2           L = A2(I,1) + 1                                                          $
2           FOR K = (2,1,L)                                                          $
2BEGIN      CA4 = CA4 + 1                                                            $
2           A4(CA4) = A2(I,K)                  END                                   $
2           FOR K = (1,1,CA4)                                                        $
2BEGIN      IF A2(A4(K),1) LSS CA4                                                   $
2           GO FIN                             END                                   $
2           B2 = 0                                                                   $
2           FOR K = (1,1,CA4)                                                        $
2BEGIN      L = A2(A4(K),1) + 1                                                      $
2           L1 = A4(K)                                                               $
2           B1 = 1                                                                   $
2           FOR K1 = (2,1,L)                                                         $
2BEGIN      FOR K2 = (1,1,CA4)                                                       $
2BEGIN      IF A2(L1,K1) EQL A4(K2)                                                  $
2           B1 = B1 + 1                        END          END                      $
2           IF B1 EQL CA4                                                            $
2           B2 = B2 + 1                        END                                   $
2           IF B2 EQL CA4                                                            $
2BEGIN      CA4 = CA4 + 1                                                            $
2           A4(CA4) = I                                                              $
2           UNTIL B2 EQL 0                                                           $
2BEGIN      R2 = A4(B2)                                                              $
2           L = A2(R2,1)                                                             $
2           IF L GEQ CA4                                                             $
```

```
2BEGIN    L1 = A2(R2,1) + 1                                                   $
2         FOR K = (2,1,L1)                                                    $
2BEGIN    R1 = A2(R2,K)                         $
2         FOR K1 = (1,1,CA4)                                                  $
2BEGIN    IF R1 EQL A4(K1)                                                    $
2         GO PP74                          END                                $
2         ENTER REMOVE                                                        $
2PP74..                                    END        END                     $
2         B2 = B2 - 1                      END                                $
2         ENTER COMBINEN                   END        END                     $
2FIN..                                     END                                $
2COMMENT  8    **************************************************
2         A PASS THROUGH THE MERGING PROCEDURES HAS NOW BEEN COMPLETED.
2         CTR IS USED TO COUNT THE NUMBER OF ROWS,REMAINING IN THE
2         MERGER DIAGRAM,WHICH HAVE NOT BEEN MERGED. IF THIS NUMBER
2         IS NOT EQUAL TO THE NUMBER STORED AS CTR1, THEN CTR1 IS
2         CHANGED TO EQUAL CTR, AND ANOTHER PASS IS MADE.
2              **************************************************     $
2         CTR = 0                                                             $
2         FOR I = (1,1,RA1)                                                   $
2BEGIN    IF (A2(I,1) GTR 0) AND (A2(I,1) LSS 1000)                           $
2         CTR = CTR + 1                    END                                $
2         IF CTR NEQ CTR1                                                     $
2BEGIN    CTR1 = CTR                                                          $
2         GO PP6                           END                                $
2COMMENT  9    **************************************************
2         IT HAS BEEN DETERMINED THAT THE LAST PASS THROUGH THE MERGING
2         PROCEDURES FAILED TO REDUCE THE THE MATRIX COMPLETELY. THIS
2         INDICATES THAT THE MERGER DIAGRAM HAS EITHER BEEN REDUCED
2         COMPLETELY, OR THAT THE GIVEN RULES HAVE NOT COMPLETELY
2         REDUCED THE MATRIX.
2         IF CTR IS NOT EQUAL TO ZERO, THEN THE DIAGRAM HAS NOT BEEN
2         COMPLETELY REDUCED. THE ROWS WHICH HAVE NOT BEEN REDUCED ARE
2         NOW PRINTED ON THE LINE PRINTER ALONG WITH AN ERROR MESSAGE.
```

```
2                ***********************************************************         $
2          IF CRT NEQ 0                                                               $
2BEGIN     WRITE($$NA)                                                                $
2FORMAT    NA(*THE PROGRAM IS UNABLE TO OFFER A SOLUTION, WITHOUT TRIAL*,
2             * AND ERROR,*,W4,*WHICH WOULD DEFINTELY BE OF MINIMUM*,
2             * FORM FOR THE FOLLOWING ROWS*,W0,
2             B5,*ROW*,W4,B6,*NO*,B7,*COMBINES WITH ROW(S) NO*,W0)                   $
2          FOR I = (1,1,RA1)                                                          $
2BEGIN     IF (A2(I,1) GTR 0) AND (A2(I,1) LSS 1000)                                  $
2          WRITE($$NA1,NA2)                  END         END                          $
2OUTPUT    NA1(I, FOR K = (2,1,10) $ A2(I,K))                                         $
2FORMAT    NA2(B5,I3,B3,10I5,W0)                                                      $
2          RA5A = RA5                                                                 $
2          FOR I = (1,1,RA1)                                                          $
2BEGIN     IF (A2(I,1) GTR 0) AND (A2(I,1) LSS 1000)                                  $
2BEGIN     RA5A = RA5A + 1                                                            $
2          A5(RA5A,1) = I                    END         END                          $
2COMMENT 10      ***********************************************************
2          THE ROWS WHICH HAVE BEEN INDICATED AS NOT OBTAINING OPTIMUM
2          MERGER ARE NOW ADDED INDIVIDUALLY TO THE LIST IN MATRIX A5(,).
2                ***********************************************************         $
2          FOR I = (1,1,RA1)                                                          $
2BEGIN     IF A2(I,1) GEQ 1000                                                        $
2BEGIN     K = A2(I,1)/1000                                                           $
2          FOR J = (1,1,RA5A)                                                         $
2BEGIN     FOR K1 = (1,1,RA1)                                                         $
2BEGIN     IF A5(J,K1) EQL K                                                          $
2BEGIN     FOR K2 = (1,1,RA1)                                                         $
2BEGIN     IF A5(J,K2) EQL 0                                                          $
2BEGIN     A5(J,K2) = I                                                               $
2          GO PP7                            END         END         END
2                                            END         END         END              $
2PP7..                                       END                                      $
2COMMENT 11      ***********************************************************
```

```
2          THE COMPLETE MERGER DIAGRAM IS NOW PRINTED ON THE LINE PRINTER
2          ACCORDING TO THE FOLLOWING TITLE AND FORMAT.
2               ***********************************************************     $
2          WRITE($$TITL5)                                                        $
2FORMAT    TITL5(*THE FOLLOWING MERGERS HAVE BEEN DETERMINED*,W4,
2                *MERGER*,W0,
2                *NUMBER   ROW(S) TO BE MERGED*,W0)                              $
2          MATRIXPRINT1(RA5A,RA1,A5(,),5)                                        $
2COMMENT   12   ************************************************************
2          THIS CONCLUDES PART4. THE REQUIRED OUTPUT OF THIS SEGMENT IS,
2          RA5A,     THE NUMBER OF ROWS TO BE COMBINED IN THE REDUCED
                    FLOW MATRIX.
2          A5(,),   THE MATRIX WHICH CONTAINS  THE ROWS TO BE MERGED AND
2                  COMBINED.
2               ***********************************************************     $
2          GO PP76                                                               $
2          END PART4                                                             $
```

Segment Five: Construction of the Reduced Matrix

The purpose of Segment Part 5 is to construct the reduced flow matrices as directed by Segment Part 4. The segment is divided into two parts, and the code HUF determines which part is used.

If HUF = 1, then the flow matrix currently under consideration is of Huffman's notation, and the first half of the segment controls the construction of the reduced flow matrix. Otherwise, HUF = 0, and the second half of the segment will control the construction of Ginsburg's reduced flow matrix.

The reduced flow matrix and a title are printed on the line printer at the conclusion of the segment, and the program transfers to the start of the master program in order to begin reduction of the next Data Set. An example of the printout of both types of reduced flow matrices is shown in Appendix B, and the program segment is shown in Figure 33.

```
2SEGMENT PART5                              $          BEGIN
2COMMENT 1      ************************************************
2        SEGMENT PART5 BEGINS HERE. THIS SEGMENT CONSTUCTS THE
2        REDUCED FLOW MATRIX AS IS SPECIFIED BY SEGMENT PART4. THE
2        BOOLEAN CODE HUF DETERMINES WHICH REDUCED FLOW MATRIX
2        IS BEING CONSTUCTED. IF HUF = 1, THEN THE MATRIX IS OF HUFFMANS
2        TYPE, AND THE FOLLOWING COMMANDS WILL CONSTUCT THE REDUCED
2        MATRIX.
2               ***********************************************     $
2PP77..  EITHER IF HUF                                              $
2BEGIN   RA5B = RA5A + 2                                            $
2        FOR I = (1,1,RA1)                                          $
2        A4(I) = 0                                                  $
2        CA4 = 0                                                    $
2        FOR I = (RA5A,-1,1)                                        $
2BEGIN   RA5B = RA5B - 1                                            $
2        FOR I1 = (1,1,CA1-1)                                       $
2BEGIN   A5(RA5B,I1) = 0                                            $
2        A5(RA5B,I1+CA1-1) = 3          END                         $
2        FOR I1 = (1,1,RA1)                                         $
2BEGIN   K = A5(I,I1)                                               $
2        IF K EQL 0                                                 $
2        GO PP81                                                    $
2        FOR I2 = (1,1,CA1-1)                                       $
2BEGIN   EITHER IF A5(RA5B,I2) GEQ 1000                             $
2BEGIN   IF A1(K,I2) GEQ 1000                                       $
2BEGIN   K1 = A1(K,I2)/1000                                         $
2        K2 = A5(RA5B,I2)/1000                                      $
2        CA4 = CA4 + 1                                              $
2        A4(CA4) = K1(10000) + K2(100) + I2                         $
2        A5(RA5B,I2 + CA1 - 1) = A1(K,CA1)      END        END       $
2        OR IF A1(K,I2) GEQ 1000                                    $
2BEGIN   A5(RA5B,I2) = A1(K,I2)                                     $
2        A5(RA5B,I2 + CA1 - 1) = A1(K,CA1)      END        END       $
```

Figure 33. ALGOL Program of Segment Part 5.

(Continued on the next 3 pages.)

```
2          IF A5(RA5B,I2) EQL 0                                                   $
2          A5(RA5B,I2) = A1(K,I2)                                                 $
2                                             END        END                      $
2PP81..                                       END                                 $
2          RA5B = RA5A + 1                                                        $
2          K1 = CA1-1                                                             $
2          K2 = CA1 + CA1 - 2                                                     $
2          FOR I = (1,1,CA4)                                                      $
2BEGIN     IF A4(I) GTR 0                                                         $
2BEGIN     K = MOD(A4(I),100)                                                     $
2          A4(I) = A4(I)/100                                                      $
2          K1 = MOD(A4(I),100)                                                    $
2          K2 = A4(I)/100                                                         $
2          FOR I1 = (1,1,RA5B)                                                    $
2BEGIN     IF A5(I1,K) EQL K2                                                     $
2          A5(I1,K) = K1                      END        END        END           $
2          K = (2)(CA1) - 2                                                       $
2          RA5B = RA5A + 1                                                        $
2          FOR I = (2,1,RA5B)                                                     $
2BEGIN     FOR I1 = (1,1,K)                                                       $
2BEGIN     A5(I-1,I1) = A5(I,I1)              END        END                      $
2          K = CA1 - 1                                                            $
2          WRITE($$TITLR)                                                         $
2FORMAT    TITLR(*HUFFMANS REDUCED MATRIX*,W4,B3,
2           *------EXCITATION--(OUTPUT)--MATRICES----------*,W0,B3,
2           *----A----   ----B----   ----C----   ----D----*,W0)                   $
2FORMAT    F1(4(I8,*(*,I2,*)*),W2)                                                $
2OUTPUT    ROW1(FOR J = (1,1, K ) $ (A5(I,J),A5(I,J+ K )) )                      $
2          FOR I = (1,1,RA5A)                                                     $
2          WRITE($$ROW1,F1)                                                       $
2                                             END                                 $
2COMMENT 2     ****************************************************
2          IF HUF = 0, THEN THE REDUCED FLOW MATRIX IS OF GINSBURGS
2          CLASS, AND THE REDUCED FLOW MATRIX IS CONSTUCTED BY THE
```

```
2          FOLLOWING COMMANDS.
2              ****************************************************          $
2          OTHERWISE                                                          $
2BEGIN     RA5B = RA5A + 2                                                    $
2          FOR I = (1,1,RA1)                                                  $
2          A4(I) = 0                                                          $
2          FOR I = (1,1,RA5A)                                                 $
2BEGIN     FOR I1 = (1,1,RA1)                                                 $
2BEGIN     K = A5(I,I1)                                                       $
2          IF K EQL 0                                                         $
2          GO TP9                                                             $
2          A4(K) = I                        END                               $
2TP9..                                      END                               $
2          FOR I = (RA5A,-1,1)                                                $
2BEGIN     RA5B = RA5B - 1                                                    $
2          FOR I1 = (1,1,CA1)                                                 $
2BEGIN     A5(RA5B,I1) = 0                                                    $
2          A5(RA5B,I1+CA1) = 0              END                               $
2          FOR I1 = (1,1,RA1)                                                 $
2BEGIN     K = A5(I,I1)                                                       $
2          IF K EQL 0                                                         $
2          GO PP91                                                            $
2          FOR I2 = (1,1,CA1)                                                 $
2BEGIN     IF A5(RA5B,I2) EQL 0                                               $
2          A5(RA5B,I2) = A1(K,I2)                                             $
2          IF A5(RA5B,I2+CA1) EQL 0                                           $
2          A5(RA5B,I2+CA1) = A10(K,I2)                END         END         $
2PP91..                                     END                               $
2          RA5B = RA5A + 1                                                    $
2          FOR I = (2,1,RA5B)                                                 $
2BEGIN     FOR I1 = (1,1,CA1)                                                 $
2BEGIN     K = A5(I,I1)                                                       $
2          IF K GTR 0                                                         $
2          A5(I,I1) = A4(K)                 END         END                   $
```

```
2           FOR I = (2,1,RA5B)                                                  $
2BEGIN      FOR I1 = (1,1,CA1+CA1)                                              $
2BEGIN      A5(I-1,I1) = A5(I,I1)            END         END                    $
2           WRITE($$TITG)                                                       $
2FORMAT     TITG(*GINSBURG REDUCED MATRIX*,W4,
2                *    --------      INPUTS------------- *,W0,
2                *    ---A---   ---B---   ---C---   ---D---*,W0)                $
2           MATRIXPRINT1(RA5A,CA1,A5(,),10)                END                  $
2           GO PP60                                                             $
2           END PART5                                                           $
2COMMENT 3       **************************************************************
2           THE REDUCED FLOW MATRIX HAS BEEN CONSTUCTED AND PRINTED ON THE
2           LINE PRINTER AT THIS POINT. THE PROGRAM NOW TRANSFERS TO THE
2           MASTER PROGRAM, AND THE NEXT FLOW MATRIX IS CONSIDERED.
2                **************************************************************  $
2FINISH                                                                         $
```

Total Reduction of a Flow Matrix

As illustrated in Chapter V and as shown by some of the examples of Appendix B, the computer program as it now stands will not always construct the minimum reduced flow matrix. This is due to the random selection of restrictions to be satisfied in the reduced flow matrix. However, modifications may be made to the program so that it will always produce the minimum attainable flow matrix through merging. The necessary changes in the analysis are as explained in Chapter V.

The major required change will be in Segment Part 3, where the basic idea as outlined in the ALGOL program will be the same, but the segment itself must be rewritten. Other than this, the only other additions will be some tests in the Main Program before each segment is overlayed. The blocks enclosed in dashed lines in Figure 34 illustrate the required changes to the segmented block diagram of Figure 22 in order to achieve the complete reduction program. It is noted from Figure 34 that additions are required to segments one and two and that segments four and five remain unchanged. The program as outlined in Figure 34 follows exactly the procedure outlined in Chapter V.

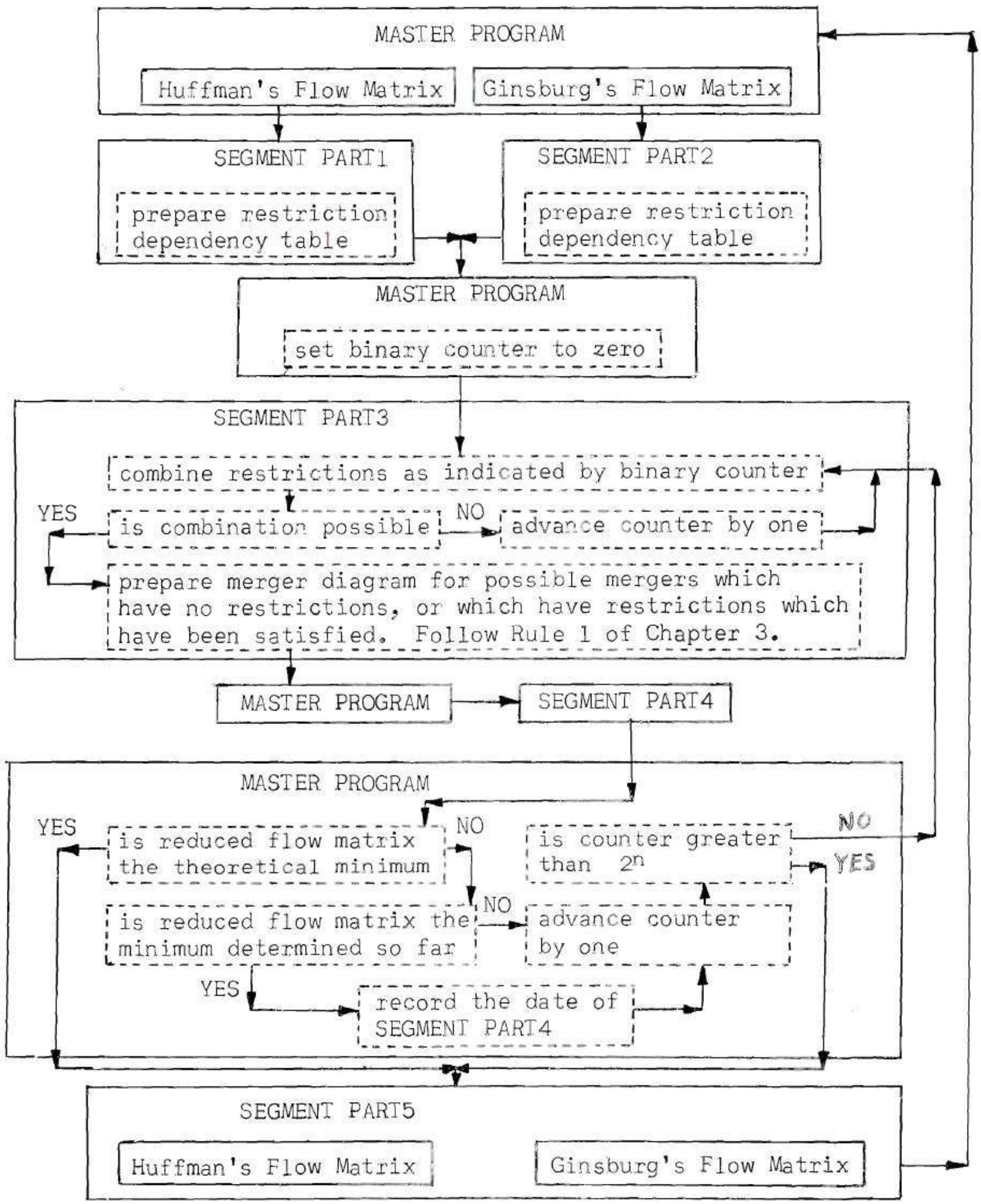

Figure 34. Segmented Block Diagram of Complete Reduction Program

CHAPTER VII

SUMMARY AND CONCLUSION

It has been the purpose of this investigation to develop a unique technique for the systematic reduction of a sequential switching system to a minimum form. It was first necessary to examine Huffman's reduction procedure, and particular attention was devoted to the "dependencies" encountered in applying the two parts of his reduction procedure. It was shown that there are dependencies among the equivalent states of Huffman's flow matrix, as well as dependencies of mergers upon the redundant states. A reduction procedure presented by Ginsburg was subsequently partially examined, and Ginsburg's idea of compatible states and determination of the theoretical minimum number of states in reduced flow matrix were presented. It was found that the idea of compatible states could also be applied to Huffman's flow matrix and the theoretical minimum number of states in his reduced flow matrix could also be determined.

Next, Ginsburg's idea of compatible states was extended so that it was possible to form a merger-restriction table. The merger-restriction table could be established for either Ginsburg's flow matrix, or Huffman's flow matrix. In either case, the merger-restriction table completely specified the flow matrix it represents in the consideration of all possible reduced flow matrices through merging.

It was found herein that the restrictions of the merger-restriction table are actually redundant states and were considered by Huffman in his reduction procedure. However, it was also found that not all redundant states of a flow matrix would be considered as restrictions.* Thus, the method of determining the restrictions automatically removes from consideration any redundant states which will not directly affect the determination of the minimum flow matrix.

Concise rules were given in Chapter IV for determining which states of the original flow matrix should be considered as a single state in the reduced flow matrix. The results of an analysis of the merger-restriction table, based upon these rules, is a merger diagram as introduced by Huffman. However, the merger diagram as introduced herein presents both a choice of states which may be merged in the reduced flow matrix, and states which must be combined in the reduced flow matrix.

Because of dependencies among the restrictions, it was necessary to establish a restriction-dependency table. Using the two tables and the associated rules, it was then shown that any desired combination of the restriction could be tested as to whether the combination would be successful. Based upon any successful combination of restrictions, it is then possible to produce the merger diagram and from this determine the minimum flow table possible based upon the desired combinations. Because of the tables and rules, it is possible to automate the procedure even for the most complicated case. An undesired feature of the procedure is the fact that some flow matrices may require the laborious consideration of all combinations of restrictions before the minimum reduced flow matrix is found. This is particularly true when the theoretical minimum flow

*This is illustrated by Example 3 of Appendix B.

matrix is unattainable, or when only one particular combination of the restrictions will yield the theoretical minimum flow matrix. Because of this possibility, a percentage reduction of the flow matrix is explained in Chapter V.

To completely automate the procedure, it was necessary to develop a systematic method for combining the states when given a merger diagram. In Chapter VI, several rules were presented for achieving this result. The rules are based upon the idea that cases which occur in a merger diagram may be removed and the one or more states involved are than combined as a single state in the reduced flow matrix. If the remaining states of the merger diagram are combined in an optimum manner, then the minimum attainable flow matrix, based upon the original merger diagram, is obtained. The rules presented are not necessarily complete; however, their application successfully reduced all merger diagrams encountered herein. If new situations are found, it is felt that new rules could be formulated which would again allow completely automatic merging.

The complete reduction procedure was programmed for a digital computer, and the resultant program is presented in Chapter VI. This program considers only one combination of restrictions for each flow matrix which it is to reduce. The restrictions to be combined are selected in a random fashion; however, the changes necessary to render the program completely automatic are outlined. Several examples of flow matrices which have been reduced are given in Appendix B.

An automatic reduction procedure has been developed by Aufenkamp; however, the method of analysis presented herein is different from his approach. As a check, one of Aufenkamp's examples is analyzed as Example Ten of Appendix B. These results were identical with those obtained by Aufenkamp. This should be expected since both procedures are designed to determine the minimum possible flow matrix through merging.

Finally the examples presented by Ginsburg in the article concerning his reduction procedure are examined by the procedure presented herein. For two of Ginsburg's examples, it is possible to determine the minimum attainable flow matrix through merging (Examples Seven and Eight of Appendix B). However, a third example presented by Ginsburg failed to be successfully reduced. This is presented as Example Nine of Appendix B. For this example, it is impossible to find the minimum flow matrix by any procedure which relies upon merging. The reduction of this flow matrix, based upon a double merging process is discussed under Example Ten, and the development of this double merging process would prove to be an interesting and useful investigation.

A P P E N D I C E S

APPENDIX A

EXAMPLE II: THE REDUCTION PROCEDURE

This appendix is devoted to presenting an illustration of a flow matrix reduction based upon the steps outlined in Chapter V. A reasonably complicated example has been chosen in order to show completely each step of the procedure as explained in the preceding chapters.

The flow matrix to be reduced is shown in Figure A1 and is also given as an exercise in Caldwell (5).

State	X_1	X_2	X_3	X_4	Z_1
1	1	11	4	10	01
2	5	2	-	3	11
3	5	2	13	3	11
4	12	-	4	15	00
5	5	-	8	-	10
6	14	6	-	10	11
7	7	6	8	3	01
8	7	-	8	3	00
9	9	11	13	10	01
10	12	6	13	10	11
11	5	11	-	3	11
12	12	2	4	15	01
13	1	-	13	10	00
14	14	-	8	-	10
15	1	6	4	15	11

Figure A1. A Flow Matrix to Be Reduced by the Reduction Procedure of Chapter V

Upon first examination, the set of possible compatible state-pairs are found and indicated in Figure A2. The numbers represent the state numbers of the flow matrix, and a check in a block indicates that the state numbers which form the row and column numbers are a possible compatible state-pair upon first examination. Figure A2 is a pictorial representation of $P_1 \equiv H \bigcup \{ \cdots \}$, as given for the previous flow matrices.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
√	√	√	√	X	√	√	√	√	√	√	√	√	X	√	1
	√	√	√	√	√	√	√	√	√	√	√	√	√	√	2
		√	√	√	√	√	√	√	√	√	√	√	√	√	3
			√	√	√	√	√	√	√	√	√	√	√	√	4
				√	√	X	√	X	√	√	X	√	√	√	5
					√	√	√	√	√	√	√	√	√	√	6
						√	√	√	√	√	√	√	X	√	7
							√	√	√	√	√	√	√	√	8
								√	√	√	√	√	X	√	9
									√	√	√	√	√	√	10
										√	√	√	√	√	11
											√	√	X	√	12
												√	√	√	13
													√	√	14
														√	15

Figure A2. Matrix Illustrating the Possible Compatible State-Pairs of Figure A1 Upon First Examination

It is now necessary to examine the matrix of Figure A2 in order to eliminate any possible compatible state-pairs which are actually incompatible. Upon first examination, it is found that several state-pairs are eliminated. That which corresponds to $P_2 \equiv H \bigcup \{ \cdots \}$ is shown in Figure A2. Again, the numbers corresponding to the row and column which contains a check are a possible compatible state-pairs under P_2.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
√			√			√	√	√	√		√			√	1
	√	√		√						√			√		2
		√		√						√			√		3
			√		√			√	√		√	√		√	4
				√	√					√			√		5
					√								√		6
						√	√								7
							√								8
								√			√	√		√	9
									√			√		√	10
										√			√		11
											√	√			12
												√		√	13
													√		14
														√	15

Figure A3. Matrix Illustrating the Possible Compatible State-Pairs of Figure A1 Upon Second Examination.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	X	X		X	X	X	X		X	X			X	X	1
			X		X	X	X	X	X		X	X		X	2
			X	X	X	X	X	X	X		X	X	X	X	3
				X	X	X	X	X		X				X	4
						X	X	X	X		X	X		X	5
						X	X	X	X	X	X	X		X	6
								X	X	X	X	X	X	X	7
								X	X	X	X	X	X	X	8
									X	X			X	X	9
										X	X		X		10
											X	X		X	11
													X	X	12
													X		13
														X	14
															15

Figure A4. Matrix Illustrating the Incompatible State-Pairs of Figure A1

No other set will contain more than five elements after each set is examined; therefore, the minimum flow matrix will theoretically contain five states.

The merger-restriction diagram for the flow matrix is shown in Figure A6, and the corresponding restriction-dependency table is shown in Figure A7. There are seven restrictions in the table; hence, there would be 128 theoretically possible reduced flow matrices.

The first reduced flow matrix will be attempted when all restrictions are satisfied. The set of combinations to be satisfied is now

$$\{(1,9),\ (1,12),\ (2,11),\ (4,13),\ (5,14),\ (9,12),\ (10,15)\}\ . \qquad (A4)$$

Set No.	States	Number of States
1	{1,2,3,5,6,7,8,10,11,14,15}	11
2	{2,4,6,7,8,9,10,12,13,15}	10
3	{3,4,5,6,7,8,9,10,12,13,14,15}	12
4	{4,5,6,7,8,11,14}	7
5	{5,7,8,9,10,12,13,15}	8
6	{6,7,8,9,10,11,12,13,15}	9
7	{7,9,10,11,12,13,14,15}	8
8	{8,9,10,11,12,13,14,15}	8
9	{9,10,14,15}	4
10	{10,11,12,14}	4
11	{11,12,13,15}	4
12	{12,14,15}	3
13	{13,14}	2
14	{14,15}	2

Figure A5. Ordering of the Incompatible States of Figure A1 to Determine the Number of States in the Theoretical Minimum Attainable Flow Matrix

Possible Mergers	Restrictions
(1,4)	(1,12),(10,15)
(1,9)	(4,13)
(1,12)	(2,11),(10,15)
(1,13)	(4,13)
(2,3)	
(2,5)	
(2,11)	
(2,14)	(5,14)
(3,11)	(2,11)
(4,9)	(9,12),(4,13),(10,15)
(4,10)	(4,13),(10,15)
(4,12)	
(4,13)	(1,12),(10,15)
(4,15)	(1,12)
(5,6)	(5,14)
(5,11)	
(5,14)	
(6,14)	
(7,8)	
(9,12)	(2,11),(4,13),(10,15)
(9,13)	(1,9)
(10,13)	(1,12)
(10,15)	(1,12),(4,13)
(11,14)	(5,14)
(12,13)	(1,12),(4,13),(10,15)
(13,15)	(4,13),(10,15)

Figure A6. Merger-Restriction Table for the Flow Matrix of Figure A1

Restrictions	Dependencies
(1,9)	(4,13)
(1,12)	(2,11),(10,15)
(2,11)	
(4,13)	(1,12),(10,15)
(5,14)	
(9,12)	(2,11),(4,13),(10,15)
(10,15)	(1,12),(4,13)

Figure A7. Restriction-Dependency Table for the Flow Matrix of Figure A1

Each state-pair must be checked separately against the partial merger diagram. In order of listing in the set given by Figure A7, the verification proceeds as follows:

(1) (1,9) is possible, and is indicated in Figure A8(a). Note that the Dependency (4,13) has also been combined.

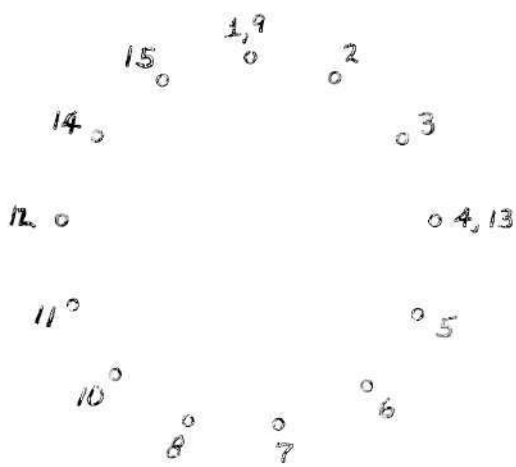

Figure A8(a). Partial Merger Diagram for Verifying Combination (1,12)

(2) (1,12) is possible only if (9,12) is possible. (9,12) is listed as a possible merger; hence its restrictions must be satisfied. Since its restrictions are included in the remaining elements which are to be verified, (9,12) and hence (1,12) will be accepted. This is indicated in Figure A8(b).

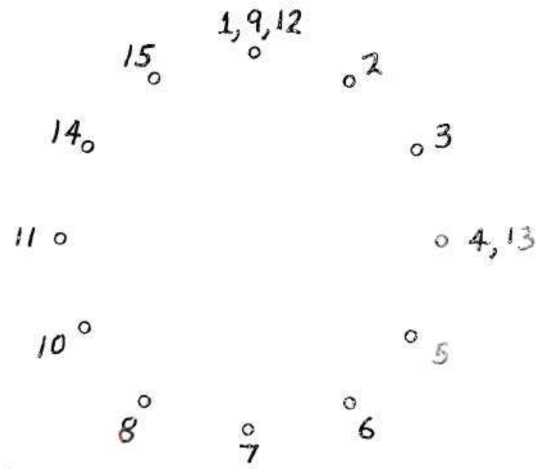

Figure A8(b). Partial Merger Diagram for Verifying Combinations (2,11),(4,13),(5,14), and (10,15)

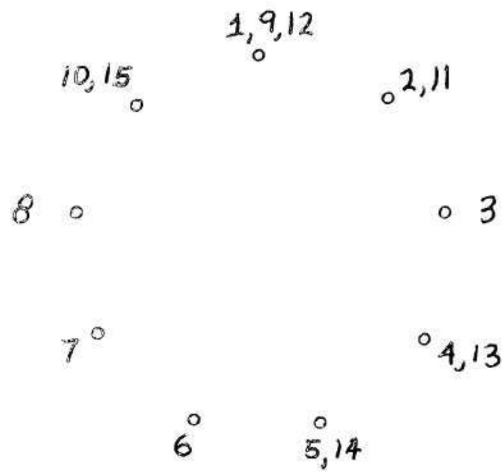

Figure A8(c). Partial Merger Diagram for Examining Mergers of Figure A6

(3) (2,11), (4,13), (5,14) and (10,15) are all accepted since their restrictions are listed in the set of combinations, and each pair satisfied the rule of combining. The partial merger diagram is completed in Figure A8(c).

Here, all combinations were accepted since all combinations satisfy the rules, and all dependencies are satisfied. The possible mergers are listed in the merger-dependency table and are each checked with the aid of Rules 1 and 2 and the partial merger diagram of Figure A8(c). The possible mergers are checked as follows:

(3) (1,4) may be indicated as a merger only if (1,13), (4,9), (4,12), (9,13) and (12,13) may be merged. Since all mergers are possible, (1,4) may be indicated as a merger in Figure A9.

(2) (2,3) may be indicated as a merger only if (3,11) may be merged. This is possible; thus, (2,3) is so indicated.

(3) (2,5) may be indicated as a merger only if (2,14), (5,11), and (11,14) may be merged. All rules are satisfied; hence (2,5) may be so indicated.

(4) (4,10) may be indicated as a merger only if (4,15), (10,13), and (13,15) may be merged. All rules are satisfied; hence, (4,10) may be indicated as a merger.

(5) (5,6) may be indicated as a merger since (6,14) may be merged.

(6) (7,8) may be indicated as a merger.

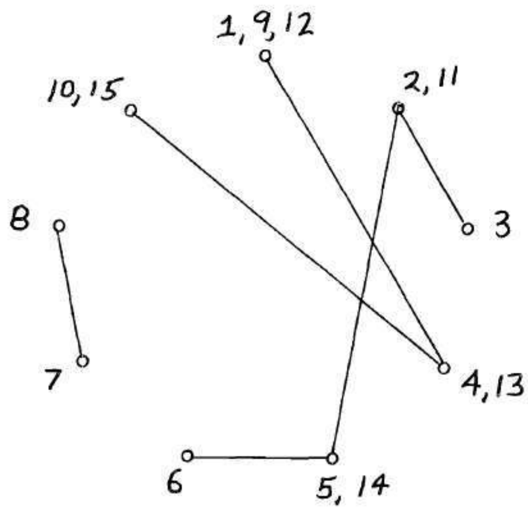

Figure A9. Merger Diagram for the Flow Matrix of Figure A1

All mergers have been indicated in the merger diagram of Figure A9. From the figure, it is seen that the reduced flow matrix will contain five states. Thus, the theoretical minimum flow matrix has been found. The excitation matrix, and the output matrix are shown in Figure A10(a) and Figure A10(b), respectively.

State	X_1	X_2	X_3	X_4
$\{1\} = \{1,4,9,12,13\}$	1	2	4	10
$\{2\} = \{2,3,11\}$	5	2	4	3
$\{3\} = \{5,6,14\}$	5	6	8	3
$\{4\} = \{7,8\}$	7	6	8	3
$\{5\} = \{10,15\}$	1	6	4	10

Figure A10(a). Excitation Matrix for the Flow Matrix of Figure A1

State	X_1	X_2	X_3	X_4
1	01	-	00	-
2	-	11	-	11
3	10	11	-	-
4	01	-	-	11
5	-	-	-	11

Figure A10(b). Output Matrix for the Flow Matrix of Figure A9(c)

APPENDIX B

AUTOMATIC REDUCTION OF FLOW MATRICES

This Appendix is devoted to examples of flow matrices which have been taken from the literature and used as input data to the reduction program described in Chapter VI. The data shown in the example are the actual output data from the program. Any important fact about an example is explained in this Appendix.

Example One

Example One is shown in Figure B1, and is the example of Figure 10 of Chapter IV. The first output from the program is the reproduction of the flow matrix to be reduced. As determined from the data of Figure 11(b), the minimum reduced flow matrix will contain two states. The next section of Figure B1 shows this fact along with a printout of the state numbers which form the set of incompatible state numbers. In the situation where more than one set contains the same number of states, as the number of states in the theoretical minimum attainable flow matrix [as illustrated by the data of Figure 11(b)], the first set is chosen for the output data. The merger-restriction table is the next output data, and for this example the merger-restriction table is the same as shown in Figure 12. This concludes the output data from Segment Part 1 of the program.

Segment Part 3 of the program constructs the merger diagram or, more appropriately, the connection matrix. The format of the connection matrix is explained in comment 35 of Figure 27 and is the computer representation of the data of Figure 14(j).

Segment Part 4 determines the optimum mergers, and the output data of this segment shows that the reduced flow matrix will contain two rows; the first row formed by merging rows 1,2,3, and 7 at the original flow matrix, while the second row is formed by merging rows 4,5,6, and 8.

Segment Part 5 produces the reduced flow matrix as illustrated in Figures 15(a) and 15(b). The output matrix is shown in parentheses, and any parentheses which contain a "3" indicate that that output is unspecified for Huffman's flow matrix.

Example Two

The flow matrix of Figure 4 is the matrix to be reduced in Example Two. Example Two is shown in Figure B2, where it is noted that the theoretical minimum attainable flow matrix will contain eight states. For the reduced flow matrix of Figure 4, it was possible to find an eight-state reduced matrix as shown by the excitation and output matrices of Figures 3(c) and 3(d), respectively.

The computer reduction of the flow matrix yielded a reduced matrix containing nine states; thus, the minimum reduced flow matrix was not found. The error is due to the fact that all restrictions were not satisfied as shown in the connection matrix; that is, restriction (6,2) was considered as a merger in the final reduction process. Reduction of this example by the method outlined in Chapter V will yield the minimum flow matrix.

MATRIX A1 HUFFMANS FLOW MATRIX

---A---	---B---	---C---	---D---	OUTPUTS
	INPUTS			
1000	3	0	7	0
2000	3	0	7	0
1	3000	6	0	1
2	4000	5	0	0
0	4	5000	8	0
0	4	6000	8	0
2	0	5	7000	1
1	0	6	8000	0

MINIMUM NUMBER OF ROWS ATTAINABLE = 2
THE ROW NUMBERS ARE 1 4

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	-----RESTRICTIONS----			
1002	0	0			
1003	0	0			
1007	1	1002	0	0	0
2003	1	1002	0	0	0
2007	0	0			
3007	2	1002	5006	0	0
3008	0	0			
4005	0	0			
4006	1	5006	0	0	0
4007	0	0			
4008	2	1002	5006	0	0
5006	0	0			
5008	1	5006	0	0	0
6008	0				

CONNECTION MATRIX

ROW NO.	NO. OF MERGERS POSSIBLE	NUMBER OF THE ROW(S) FOR WHICH A MERGER IS POSSIBLE						
1	2	3	7	0	0	0	0	0
2	ROW 2 MUST MERGE WITH ROW 1							
3	3	1	7	8	0	0	0	0
4	3	5	7	8	0	0	0	0
5	2	4	8	0	0	0	0	0
6	ROW 6 MUST MERGE WITH ROW 5							
7	3	1	3	4	0	0	0	0
8	3	3	4	5	0	0	0	0

Figure B1. Data of Example One.
(Continued next page.)

```
THE FOLLOWING MERGERS HAVE BEEN DETERMINED
MERGER
NUMBER    ROW(S) TO BE MERGED
     1        3    7    1    2    0    0    0    0
     2        5    8    4    6    0    0    0    0

HUFFMANS REDUCED MATRIX
*  ------EXCITATION--(OUTPUT)--MATRICES---------
*  ----A----    ----B----    ----C----    ----D----
   1000( 0)     3000( 1)        5( 3)     7000( 1)
      1( 3)     4000( 0)     5000( 0)     8000( 0)
```

MATRIX A1 HUFFMANS FLOW MATRIX

---A---	---B---	---C---	---D---	OUTPUTS
1000	7	9	4	11
2000	5	3	4	1
1	7	3000	11	10
2	0	3	4000	0
6	5000	9	0	11
6000	5	3	11	1
1	7000	14	0	10
8000	12	3	4	1
1	7	9000	13	1
1	7	10000	4	10
8	0	10	11000	0
6	12000	9	0	11
8	0	14	13000	11
2	12	14000	11	0

MINIMUM NUMBER OF ROWS ATTAINABLE = 8

THE ROW NUMBERS ARE 1 2 3 5 7 9 13 14

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	----RESTRICTIONS----			
2004	0	0			
2006	1	4011	0	0	0
2008	1	5012	0	0	0
2011	3	2008	3010	4011	0
3010	1	4011	0	0	0
4006	2	2006	4011	0	0
4008	1	2008	0	0	0
4011	2	2008	3010	0	0
5012	0	0			
6008	2	5012	4011	0	0
6011	2	6008	3010	0	0
8011	2	3010	4011	0	0

Figure B2. Data of Example Two.
(Continued next page.)

```
CONNECTION MATRIX
ROW   NO. OF MERGERS    NUMBER OF THE ROW(S) FOR
NO.      POSSIBLE       WHICH A MERGER IS POSSIBLE
 1           0             0
 2           2             4     6     0     0     0     0
 3           0             0
 4           1             2     0     0     0
 5           0             0
 6           1             2     0     0     0
 7           0             0     -
 8     ROW   8   MUST MERGE WITH ROW   2
 9           0             0
10     ROW  10   MUST MERGE WITH ROW   3
11     ROW  11   MUST MERGE WITH ROW   4
12     ROW  12   MUST MERGE WITH ROW   5
13           0             0
14           0             0

THE FOLLOWING MERGERS HAVE BEEN DETERMINED
MERGER
NUMBER    ROW(S) TO BE MERGED
     1         1    0    0    0    -
     2         3   10    0    0    0
     3         5   12    0    0    0    0
     4         7    0    0    0    0
     5         9    0    0    0    0
     6        13    0    0    0    0
     7        14    0    0    0    0
     8         4   11    0    0    0    0
     9         2    6    8    0    0    0

HUFFMANS REDUCED MATRIX
.   ------EXCITATION--(OUTPUT)--MATRICES--------
.   -----A-----      -----B-----      -----C-----      -----D-----
     1000(11)            7( 3)            9( 3)            4( 3)
        1( 3)            7( 3)         3000(10)            4( 3)
        2( 3)         5000(11)            9( 3)            0( 3)
        1( 3)         7000(10)           14( 3)            0( 3)
        1( 3)            7( 3)         9000( 1)           13( 3)
        2( 3)            0( 3)           14( 3)        13000(11)
        2( 3)            5( 3)        14000( 0)            4( 3)
        2( 3)            0( 3)            3( 3)         4000( 0)
     2000( 1)            5( 3)            3( 3)            4( 3)
```

Example Three

The flow matrix of Figure 6(a) is considered in Example Three of Figure B3. This flow matrix was used in Chapter II to illustrate dependencies which exist among the redundant states of Huffman's flow matrix. It is used here to illustrate more clearly the relationship between the restrictions of the merger-restriction table and Huffman's redundant states.

The redundant states of this example were determined to be (1,6), (2,6), and (2,8); however, the restrictions as shown in Figure B3 are only (1,6) and (2,8). This is because none of the possible mergers depended upon the combinations of state-pair (2,6) before they may be merged. Thus, even though state-pair (2,6) satisfies the requirement of redundant states, it may be considered only as a possible merger in the analysis of the flow matrix. Such situations will always be revealed by the merger-restriction table.

The reduced flow matrix of this example will contain a minimum of three states; however, the minimum matrix found was a four-state matrix as shown in Figures 6(e) and B3.

Examples Four, Five, and Six

Examples Four, Five, and Six are shown in Figures B4, B5, and B6, respectively. There is nothing unique about any of these examples, and they are presented here only as illustrations of the automatic reduction of flow matrices.

MATRIX A1 HUFFMANS FLOW MATRIX

.	----------INPUTS----------			OUTPUTS
. --A--	--B--	--C--	--D--	--------
1000	0	4	7	10
2000	3	5	0	10
8	3000	5	0	11
	9	4000	7	1
	3	5000	7	1
6000	3	0	7	10
1	0	4	7000	0
8000	0	5	10	10
6	9000	4	0	0
2	0	5	10000	11

MINIMUM NUMBER OF ROWS ATTAINABLE = 3
THE ROW NUMBERS ARE 3 4 6

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	-----RESTRICTIONS-----			
1004	0	0			
1006	0	0			
1007	0	0			
1009	1	1006	0	0	0
2003	1	2008	0	0	0
2005	0	0			
2006	0	0			
2008	0	0			
2010	0				
3005	0				
3008	0				
3010	1	2008	0	0	0
4007	0	0			
4009	0	0			
5006	0				
6007	1	1006	0	0	0
7009	1	1006	0	0	0
8010	1	2008	0	0	0

Figure B3. Data of Example Three.
(Continued next page.)

```
CONNECTION MATRIX
ROW   NO. OF MERGERS    NUMBER OF THE ROW(S) FOR
NO.      POSSIBLE       WHICH A MERGER IS POSSIBLE
 1           1             7    0    0    0    0
 2           2             3   10    0    0    0    0
 3           3             2    5   10    0    0    0    0
 4           2             7    9    0    0    0    0
 5           1             3    0    0    0    0
 6     ROW   6   MUST MERGE WITH ROW   1
 7           3             1    4    9    0    0    0    0
 8     ROW   8   MUST MERGE WITH ROW   2
 9           2             4    7    0    0    0    0
10           2             2    3    0    0    0    0

THE FOLLOWING MERGERS HAVE BEEN DETERMINED
MERGER
NUMBER    ROW(S) TO BE MERGED
     1         5    0    0    0    0
     2         3   10    2    8    0    0    0    0
     3         1    6    0    0    0    0
     4         7    9    4    0    0    0    0

HUFFMANS REDUCED MATRIX
.    ------EXCITATION--(OUTPUT)--MATRICES--------
.    ----A----     ----B----     ----C----     ----D----
          ( 3)         3( 3)     5000( 1)         7( 3)
      2000(10)      3000(11)        5( 3)     10000(11)
      1000(10)         3( 3)        4( 3)         7( 3)
         1( 3)      9000( 0)     4000( 1)      7000( 0)
```

MATRIX A1 HUFFMANS FLOW MATRIX

INPUTS				OUTPUTS
A	B	C	D	
1000	11	4	10	1
5	2000	0	3	11
5	2	13	3000	11
12	0	4000	15	0
5000	0	8	0	10
14	6000	0	10	11
7000	6	8	3	1
7	0	8000	3	0
9000	11	13	10	1
12	6	13	10000	11
5	11000	0	3	11
12000	2	4	15	1
1	0	13000	10	0
14000	0	8	0	10
1	6	4	15000	11

MINIMUM NUMBER OF ROWS ATTAINABLE = 5

THE ROW NUMBERS ARE 1 2 6 7 10

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	RESTRICTIONS			
1004	2	1012	10015	0	0
1009	1	4013	0	0	0
1012	2	2011	10015	0	0
1013	1	4013	0	0	0
2003	0				
2005	0				
2011	0				
2014	1	5014	0	0	0
3011	1	2011	0	0	0
4009	3	9012	4013	10015	0
4010	2	4013	10015	0	0
4012	0				
4013	2	1012	10015	0	0
4015	1	1012	0	0	0
5006	1	5014	0	0	0
5011	0	*			
5014	0	*			
6014	0	0			
7008	0				
9012	3	2011	4013	10015	0
9013	1	1009	0	0	0
10013	1	1012	0	0	0
10015	2	1012	4013	0	0
11014	1	5014	0	0	0
12013	3	1012	4013	10015	0
13015	2	4013	10015	0	0

Figure B4. Data of Example Four.
(Continued next page.)

CONNECTION MATRIX

ROW NO.	NO. OF MERGERS POSSIBLE	NUMBER OF THE ROW(S) FOR WHICH A MERGER IS POSSIBLE					
1	2	4	9	0	0	0	0
2	2	3	5	0	0	0	0
3	1	2	0	0	0	0	
4	2	1	10	0	0	0	0
5	2	2	6	0	0	0	0
6	1	5	0	0	0	0	
7	1	8	0	0	0	0	
8	1	7	0	0	0	0	
9	1	1	0	0	0	0	
10	1	4	0	0	0	0	

11 ROW 11 MUST MERGE WITH ROW 2
12 ROW 12 MUST MERGE WITH ROW 1
13 ROW 13 MUST MERGE WITH ROW 4
14 ROW 14 MUST MERGE WITH ROW 5
15 ROW 15 MUST MERGE WITH ROW 10

THE FOLLOWING MERGERS HAVE BEEN DETERMINED

MERGER NUMBER	ROW(S) TO BE MERGED						
1	3	2	11	0	0	0	
2	5	6	14	0	0	0	0
3	7	8	0	0	0	0	
4	9	1	12	0	0	0	0
5	4	10	13	15	0	0	0

HUFFMANS REDUCED MATRIX

. ------EXCITATION--(OUTPUT)--MATRICES--------

. ----A----	----B----	----C----	----D----
5(3)	2000(11)	4(3)	3000(11)
5000(10)	6000(11)	8(3)	10(3)
7000(1)	6(3)	8000(0)	3(3)
9000(1)	2(3)	4(3)	10(3)
9(3)	6(3)	4000(0)	10000(11)

MATRIX A1 HUFFMANS FLOW MATRIX

INPUTS A	INPUTS B	INPUTS C	INPUTS D	OUTPUTS
1000	3	0	8	1
2000	3	0	8	10
1	3000	6	0	1
2	4000	6	0	10
	3	5000	7	1
	4	6000	8	10
1	0	5	7000	1
2	0	5	8000	10

MINIMUM NUMBER OF ROWS ATTAINABLE = 4
THE ROW NUMBERS ARE 1 2 4 5

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	RESTRICTIONS
1003	0	0
2008	0	0
4006	0	0
5007	0	*

CONNECTION MATRIX

ROW NO.	NO. OF MERGERS POSSIBLE	NUMBER OF THE ROW(S) FOR WHICH A MERGER IS POSSIBLE					
1	1	3	0	0	0	0	0
2	1	8	0	0	0	0	0
3	1	1	0	0	0	0	0
4	1	6	0	0	0	0	0
5	1	7	0	0	0	0	0
6	1	4	0	0	0	0	0
7	1	5	0	0	0	0	0
8	1	2	0	0	0	0	0

THE FOLLOWING MERGERS HAVE BEEN DETERMINED

MERGER NUMBER	ROW(S) TO BE MERGED					
1	1	3	0	0	0	0
2	2	8	0	0	0	0
3	4	6	0	0	0	0
4	5	7	0	0	0	0

HUFFMANS REDUCED MATRIX

EXCITATION--(OUTPUT)--MATRICES A	B	C	D
1000(1)	3000(1)	6(3)	8(3)
2000(10)	3(3)	5(3)	8000(10)
2(3)	4000(10)	6000(10)	8(3)
1(3)	3(3)	5000(1)	7000(1)

Figure B5. Data of Example Five

MATRIX A1 HUFFMANS FLOW MATRIX

•	--A--	--B--	--C--	--D--	OUTPUTS
	1000	3	0	6	1
•	0	3	2000	6	10
	1	3000	5	0	1
	4000	3	0	6	10
•	0	3	5000	6	1
•	4	0	2	6000	10

MINIMUM NUMBER OF ROWS ATTAINABLE - 2
THE ROW NUMBERS ARE 1 4

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	-----RESTRICTIONS----
1002	0	
1003	0	
1005	0	
2004	0	
2006	0	
3005	0	
4005	0	
4006	0	

CONNECTION MATRIX

ROW NO.	NO. OF MERGERS POSSIBLE	NUMBER OF THE ROW(S) FOR WHICH A MERGER IS POSSIBLE				
1	3	2	3	5	0	0
2	3	1	4	6	0	0
3	2	1	5	0	0	0
4	3	2	5	6	0	0
5	3	1	3	4	0	0
6	2	2	4	0	0	0

THE FOLLOWING MERGERS HAVE BEEN DETERMINED

MERGER NUMBER	ROW(S) TO BE MERGED					
1	1	5	3	0	0	0
2	2	6	4	0	0	0

HUFFMANS REDUCED MATRIX

•	------EXCITATION--(OUTPUT)--MATRICES---------			
•	----A----	----B----	----C----	----D----
	1000(1)	3000(1)	5000(1)	6(3)
	4000(10)	3(3)	2000(10)	6000(10)

Figure B6. Data of Example Six

Example Seven

Example Seven is shown in Figure B7, and is the example of Figure 15 from Ginsburg's article (11). In his article, Ginsburg introduces a new method for reducing flow matrices, and gives three examples. These examples are repeated here as Examples Seven, Eight, and Nine. Ginsburg's reduction technique is discussed and compared to the automatic reduction procedure introduced in this paper.

Example Seven is presented first since the theoretical minimum flow matrix is obtained both by Ginsburg's method, and the reduction procedure discussed in Chapters IV and V. The important fact which the author feels should be presented is that the merger-restriction table appears to be normal; that is, restrictions may be combined, and the combinational table established which yields the minimum flow matrix.

Example Eight

Example Eight of Figure B8 is the Example Three of Ginsburg's article (11). This example is discussed in detail in Chapters III and V with illustrations in Figures 8, 12, 16, 17, 19, 20, and 21. The minimum reduced flow matrix was found to be a five-state matrix as shown in Figure 21, although the theoretical minimum flow matrix was a four-state matrix. For the data in Figure B8, it is seen that the reduction program found a seven-state reduced matrix. This is because of the fact that the proper restrictions were not satisfied in the reduced flow matrix. This is the same situation that exists in Example Two explained previously.

In his article, Ginsburg also failed to find a four-state reduced matrix, and in turn, finds a five-state flow matrix both by merging and

```
MATRIX A1  GINSBURG STATE IDENIFICATION
    -----------INPUTS-----------
    --A--    --B--    --C--    --D--
        0        0        6
        0        5        0
        0        0        4
        0        0        3
        2        0        0
        0        1        0

MATRIX A10   GINSBURG OUTPUT IDENIFICATION
    -----------OUTPUTS-----------
    --------FOR INPUTS-----------
    --A--    --B--    --C--    --D--
        1        0        3
        0        2        0
        1        0        0
        0        1        0
        2        0        1
                 1        2

MINIMUM NUMBER OF ROWS ATTAINABLE -  3
THE ROW NUMBERS ARE    1    5    6

MATRIX A9-RESTRICTION MATRIX
POSSIBLE     NUMBER  OF
 MERGERS     RESTRICTIONS      -----RESTRICTIONS----
   1002           0
   1003           1            4006          0          0
   1004           1            3006          0          0
   2003           0               0
   2005           0               0
   3004           0
   3006           0
   4005           0
   4006           0
```

Figure B7. Data of Example Seven.
(Continued next page.)

```
CONNECTION MATRIX
ROW   NO. OF MERGERS    NUMBER OF THE ROW(S) FOR
NO.       POSSIBLE      WHICH A MERGER IS POSSIBLE
 1           2           2     3     0     0     0
 2           3           1     3     5     0     0
 3           3           1     2     4     0     0
 4           1           3     0     0     0     0
 5           1           2     0     0     0     0
 6     ROW   6   MUST MERGE WITH ROW   4

THE FOLLOWING MERGERS HAVE BEEN DETERMINED
MERGER
NUMBER    ROW(S) TO BE MERGED
     1       4     6     0     0     0     0
     2       5     0     0     0     0     0
     3       2     3     1     0     0     0

GINSBURG REDUCED MATRIX
.    ----------- INPUTS---------- *
.    ---A--     ---B--     ---C--     ---D--
.    0( 0)      3( 1)      3( 2)      (   )
     3( 2)      0( 0)      0( 1)      (   )
.    0( 1)      2( 2)      1( 3)      (   )
```

```
MATRIX A1  GINSBURG STATE IDENIFICATION
.   -----------INPUTS-----------
.   --A--     --B--     --C--     --D--
        2         0         3
.       0         4         0
        7         0         6
.       0         2         0
        5         0         3
.       0         1         7
        6         0         1
.       0         4         2

MATRIX A10   GINSBURG OUTPUT IDENIFICATION
.   -----------OUTPUTS----------
.   --------FOR INPUTS----------
.   --A--     --B--     --C--     --D--
        1         0         2
.       0         2         0
        2         0         1
.       0         1         0
        1         0         2
.       0         2         1
        2         0         2
.       0         2         1

MINIMUM NUMBER OF ROWS ATTAINABLE -  4
THE ROW NUMBERS ARE    1    3    6    7

MATRIX A9-RESTRICTION MATRIX
POSSIBLE    NUMBER  OF
 MERGERS    RESTRICTIONS     -----RESTRICTIONS----
   1002          0
   1004          0
   1005          1           2005       0       0       0
   2003          0
   2005          0
   2006          1           1004       0       0       0
   2007          0
   2008          0
   3004          0
   3008          1           2006       0       0       0
   4005          0
   4007          0
   6008          2           1004    2007       0       0
```

Figure B8. Data of Example Eight.
(Continued next page.)

CONNECTION MATRIX

ROW NO.	NO. OF MERGERS POSSIBLE	NUMBER OF THE ROW(S) FOR WHICH A MERGER IS POSSIBLE						
1	0	0						
2	5	3	5	6	7	8	0	0
3	1	2	0	0	0	0	0	
4	ROW 4 MUST MERGE WITH ROW 1							
5	1	2	0	0	0	0	0	
6	1	2	0	0	0	0	0	
7	1	2	0	0	0	0	0	
8	1	2	0	0	0	0	0	

THE FOLLOWING MERGERS HAVE BEEN DETERMINED

MERGER NUMBER	ROW(S) TO BE MERGED						
1	1	4	0	0	0	0	
2	3	0	0	0	0		
3	5	0	0	0	0		
4	6	0	0	0	0	0	
5	7	0	0	0	0		
6	2	8	0	0	0	0	0

GINSBURG REDUCED MATRIX

•	----------- INPUTS---------			
•	---A--	---B--	---C--	---D--
	6(1)	6(1)	2(2)	( )
	5(2)	0(0)	4(1)	( )
	3(1)	0(0)	2(2)	( )
•	0(0)	1(2)	5(1)	( )
	4(2)	0(0)	1(2)	( )
•	0(0)	1(2)	6(1)	( )

by his own procedure. Thus, the minimum-state matrix was found by either method.

Example Nine

Example Nine is important because it is one example where Ginsburg's method succeeds, and the merging techniques fail to find a reduced flow matrix. Example Nine is taken from Figure 2 of Ginsburg's article, and the computer output is shown in Figure B9.

The theoretical minimum reduced flow matrix will contain four states vs. six states in the original flow matrix. Ginsburg's procedure does in fact find a four-state reduced matrix and Ginsburg notes that the well-known merging technique yields no reduction in the flow matrix. It is noted also that the method of this article did indeed result in no reduction of the flow matrix as illustrated by the data of Figure B9.

It is necessary, however, to examine the restrictions of the connection matrix of Figure B9, where it is noted that each of the possible mergers are in turn also a restriction. Because of the way the states are paired, it is impossible to combine or merge any pairs.

It is now possible to consider a double merging process; that is, one state will be merged simultaneously with different states even though the two states themselves are not merged or combined. The possibilities which exist, as determined from the merger-restriction table, are as follows:

(1) State 1 must merge with either or both states 2 and 3, but states 2 and 3 will not merge.

```
MATRIX A1  GINSBURG STATE IDENIFICATION
.   ----------INPUTS----------
.   --A--    --B--    --C--    --D--
        5        0        4
        6        6        0
        4        4        0
        2        0        2
        1        0        0
        3        0        0

MATRIX A10   GINSBURG OUTPUT IDENIFICATION
.   ----------OUTPUTS----------
.   --------FOR INPUTS---------
.   --A--    --B--    --C--    --D--
        3        0        3
        3        1        0
        3        2        0
        4        0        1
        4        0        0
        4        0        0

MINIMUM NUMBER OF ROWS ATTAINABLE =  4
THE ROW NUMBERS ARE   2   3   4   6

MATRIX A9-RESTRICTION MATRIX
POSSIBLE     NUMBER  OF
 MERGERS    RESTRICTIONS       -----RESTRICTIONS----
   1002           1            5006        0        0
   1003           1            4005        0        0
   4005           1            1002        0        0
   5006           1            1003        0        0

CONNECTION MATRIX
ROW  NO. OF MERGERS   NUMBER OF THE ROW(S) FOR
NO.     POSSIBLE      WHICH A MERGER IS POSSIBLE
 1          0          0    0
 2          0          0
 3          0          0    0
 4          0          0
 5          0          0    0
 6          0          0
```

Figure B9. Data of Example Nine.
(Continued next page.)

THE FOLLOWING MERGERS HAVE BEEN DETERMINED

MERGER NUMBER	ROW(S) TO BE MERGED					
1	1	0	0	0	0	0
2	2	0	0	0	0	0
3	3	0	0	0	0	0
4	4	0	0	0	0	0
5	5	0	0	0	0	0
6	6	0	0	0	0	0

GINSBURG REDUCED MATRIX

•	------------ INPUTS----------				
•	---A--	---B--	---C--	---D--	
	5(3)	0(0)	4(3)	( )	( )
	6(3)	6(1)	0(0)	( )	( )
	4(3)	4(2)	0(0)	( )	( )
	2(4)	0(0)	2(1)	( )	( )
	1(4)	0(0)	0(0)	( )	( )
	3(4)	0(0)	0(0)	( )	( )

(2) State 5 must merge with either or both states 4 and 6, but states 4 and 6 will not merge.

These same two possibilities were determined by Ginsburg in his article, and from this, he proceeded with enumeration and selection of alternate paths and found a solution which contains four states. In order to accomplish this enumeration, it was necessary to examine each possibility both with respect to each input, and the required output associated with each input.

An automatic procedure for accomplishing a reduction by Ginsburg's procedure has not yet been developed; however, it may be possible to develop such a procedure based upon double merging. The merger-restriction table idea presented herein easily determines what possibilities exist; however, it will be necessary to develop some new types of tables in order to completely examine all possibilities systematically. Such an investigation would be an interesting extension of this dissertation.

Example Ten

Example Ten is taken from an article by Aufenkamp and Hohn(7). In the article, the authors consider an automatic reduction technique based upon the equivalence of the states of the flow matrix. Rules are given, and an example is given of the reduction procedure. The authors state that the process is not difficult to execute, even in the most complex cases, and could be programmed for a computer. Thus it is important that the example given by the authors be reduced by the procedure developed within this thesis, and the results of the two different procedures to be compared.

It is necessary that the flow matrix notation given by Aufenkamp and Hohn (7) be organized into a format suitable for the reduction program of this thesis. To do this, the flow matrix of Aufenkamp is converted to the flow matrix representation of Ginsburg, as described herein in Chapter III. Figure B10 illustrates the transposed example with the same output notation as used by Aufenkamp and Hohn in their paper.

State	X_1	X_2	X_3	X_4	X_5	X_6
1	9(α)	3(γ)	9(β)	-	2(δ)	-
2	(β)	11(γ)	9(β)	10(α)	3(β)	-
3	9(α)	(γ)	11(β)	-	1(δ)	-
4	1(α)	(γ)	11(β)	-	9(δ)	-
5	7(β)	5(γ)	13(β)	3(α)	4(β)	-
6	(β)	10(δ)	-	12(α)	(α)	3(γ)
7	10(α)	2(γ)	1(β)	-	1(δ)	-
8	2(β)	11(γ)	1(β)	7(α)	3(β)	-
9	1(α)	3(γ)	9(β)	-	8(δ)	-
10	10(α)	8(γ)	9(β)	-	1(δ)	-
11	2(β)	2(γ)	1(β)	7(α)	12(β)	-
12	(α)	4(γ)	9(β)	-	2(δ)	--
13	(β)	7(δ)	-	9(α)	12(α)	3(γ)

Figure B10. The Transposed Flow Matrix of the Example of Aufenkamp and Hohn (7)

Since the output designations of Figure B10 are symbols, it is necessary to provide integer values for the output symbols as follows

$$\begin{aligned} \alpha &= 1 \\ \beta &= 2 \\ \gamma &= 3 \\ \delta &= 4 \end{aligned} \tag{B1}$$

MATRIX A1 GINSBURG STATE IDENIFICATION

-----------INPUTS-----------				
--A--	--B--	--C--	--D--	
9	3	9	0	2
0	11	9	10	3
9	0	11	0	1
1	0	11	0	9
7	5	13	3	4
0	10	0	12	0
10	2	1	0	1
2	11	1	7	3
1	3	9	0	8
10	8	9	0	1
2	2	1	7	12
	4	9	0	2
0	7	0	9	12

MATRIX A10 GINSBURG OUTPUT IDENIFICATION

-----------OUTPUTS-----------				
--------FOR INPUTS----------				
--A--	--B--	--C--	--D--	
1	3	2	0	4
2	3	2	1	2
1	3	2	0	4
1	3	2	0	4
2	3	2	1	2
2	4	0	1	1
1	3	2	0	4
2	3	2	1	2
1	3	2	0	4
1	3	2	0	4
2	3	2	1	2
1	3	2	0	4
2	4	0	1	1

MINIMUM NUMBER OF ROWS ATTAINABLE - 7
THE ROW NUMBERS ARE 1 2 3 5 6 7 11

MATRIX A9-RESTRICTION MATRIX

POSSIBLE MERGERS	NUMBER OF RESTRICTIONS	-----RESTRICTIONS----			
1009	1	2008	0	0	0
1012	1	3004	0	0	0
2008	2	1009	7010	0	0
3004	2	1009	1009	0	0
6013	2	7010	9012	0	0
7010	2	2008	1009	0	0
9012	2	3004	2008	0	0

Figure B11. Data of Example Ten

Also, since input X_6 obviously will not prevent the merging or combination of any two rows, it is possible to consider only the first five inputs in the computer reduction program. This is done in the interest of space in the program.

Figure B11 shows a computer printout of the flow matrix of Figure B10, along with the merger-restriction of the flow matrix. This data will now be analyzed to determine if the reduction procedure will yield a seven-state reduced flow matrix. A seven-state flow matrix is the theoretical minimum flow matrix, and a seven-state reduced flow matrix was found by Aufenkamp and Hohn (7).

From the merger-restriction table of Figure B11, it is possible to construct the restriction table of Figure B12. This must now be analyzed as to the possible combination of restrictions, and the first trial will be the satisfaction of all restrictions.

Restrictions	Dependencies
(1,9)	(2,8)
(2,8)	(1,9), (7,10)
(3,4)	(1,9)
(7,10)	(2,8), (1,9)
(9,12)	(3,4), (2,8)

Figure B12. Restriction Dependency Table for the Flow Matrix of Figure B10

Examination of the restrictions according to the established rules proves that all restrictions may be satisfied; thus, the modified merger diagram is as shown in Figure B13. It is now necessary to examine the merger-restriction table for possible mergers. There is only one possible merger to be examined namely (6,13), and it satisfies all the rules. The final merger diagram is then shown in Figure B14.

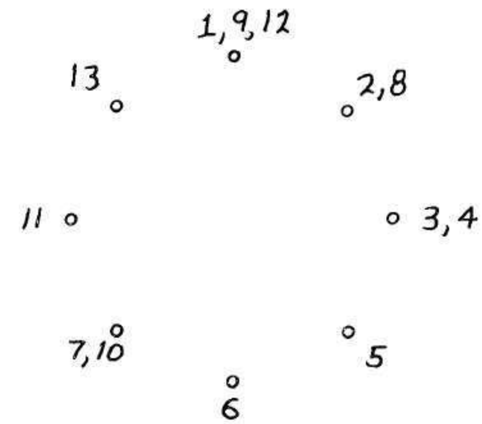

Figure B13. Partial Merger Diagram for the Examination of Possible Mergers in Merger-Restriction Table of Figure B11

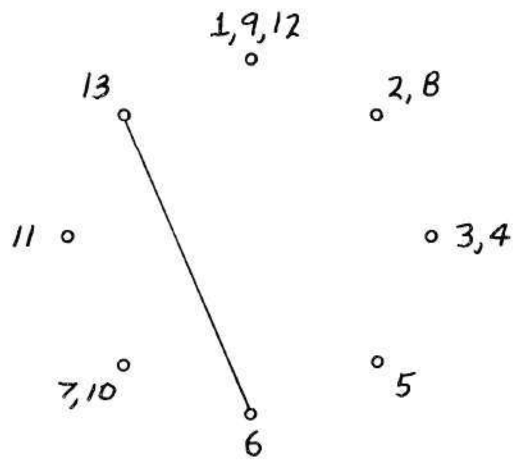

Figure B14. Merger Diagram for the Flow Matrix of Figure B10

It is seen from Figure B14 that the reduced flow matrix will be a seven-state flow matrix; thus, the theoretical minimum flow matrix has been found. The reduced flow matrix is shown in Figure B15.

As would be suspected, the reduced flow matrix determined in Example Ten is exactly the same as determined by Aufenkamp and Hohn (7). This is true since both procedures start with the same example and try to find the minimum flow matrix attainable; however, the two procedures consider a different approach to the reduction problem.

Aufenkamp's and Hohn's procedure consists of partitioning the states of the flow matrix according to their compatibility. The partitioning process is repeated in a manner similar to the procedure described previously for determining compatible state-pairs, except that the flow matrix is considered as sets of states rather than sets of state-pairs as described herein. Aufenkamp's procedure will develop a number of different partitionings which may be compared in order to determine which partitioning yields the minimum flow matrix.

State	X_1	X_2	X_3	X_4	X_5	X_6
$\{1\} = \{1,9,12\}$	(α)	$3(\gamma)$	(β)	-	$2(\delta)$	-
$\{2\} = \{2,8\}$	(β)	$7(\gamma)$	$1(\beta)$	$6(\alpha)$	$3(\beta)$	-
$\{3\} = \{3,4\}$	$1(\alpha)$	(γ)	$7(\beta)$	-	$1(\delta)$	-
$\{4\} = \{5\}$	$6(\beta)$	(γ)	$6(\beta)$	$3(\alpha)$	$3(\beta)$	-
$\{5\} = \{6,13\}$	(β)	$6(\delta)$	-	$1(\alpha)$	$1(\alpha)$	$3(\gamma)$
$\{6\} = \{7,10\}$	(α)	$2(\gamma)$	$1(\beta)$	-	$1(\delta)$	-
$\{7\} = \{11\}$	$2(\beta)$	$2(\gamma)$	$1(\beta)$	$6(\alpha)$	$1(\beta)$	-

Figure B15. The Reduced Flow Matrix of Figure B10

APPENDIX C

UNSPECIFIED OUTPUTS FOR HUFFMAN'S FLOW MATRIX

One author, Marcus (14), considers the reduction of a flow matrix of Huffman's type when the outputs are not completely specified (14). A modification of the rules established for determining the merger-restriction table will result in rules for establishing the merger-restriction table for a flow matrix with unspecified outputs.

In Marcus' example, the state output of a stable state was unspecified; however, a more general case will be considered here. That is, how would the merger-restriction table be established for a flow matrix in which part or all of any output is unspecified. To do this, it is necessary to separate the outputs into their separate identities.

As an example of unspecified outputs, consider the flow matrix of Figure B3 of Appendix B. The outputs of the flow matrix are separated and respecified in Figure C1.

The rules, as outlined in Chapter V for determining possible compatible state-pairs for Huffman's flow matrix, must now be rewritten. Two states will form a possible state-pair if for each output column,

(1) Each state has the same output, or

(2) One or both states have a blank entry, or

(3) Each state has a different output, but both states do not simultaneously have stable states occurring within the same input column. With the exception of the change in rules for determining possible

compatible state-pairs, the reduction procedure is exactly as outlined in the previous chapters.

X_1	X_2	X_3	X_4	Z_1	Z_2
1	-	4	7	1	0
2	3	5	-	1	0
8	3	5	-	1	-
-	9	4	7	0	1
-	3	5	7	-	-
6	3	0	7	1	0
1	-	4	7	0	0
8	-	5	10	-	0
6	9	4	-	0	0
2	0	5	10	1	1

Figure C1. A Flow Matrix with Unspecified Outputs

BIBLIOGRAPHY

1. Karnaugh, M., "The Map Method for Synthesis of Combinational Logic Circuits," Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics, Vol 72, pp. 593-599, November 1953.

2. Keister, W., Ritchie, S. A., and Washburn, S., The Design of Switching Circuits, D. Van Nostrand Co., Inc., New York, N. Y., 1951.

3. Huffman, D. A., "The Synthesis of Sequential Switching Circuits," Journal of the Franklin Institute, Vol. 257, pp. 161-190, 257-303, March/April, 1954.

4. Moore, E. F., "Gedanken-Experiments on Sequential Machines," Automata Studies, Princeton University Press, Princeton, N. J., pp. 129-153, 1956.

5. Caldwell, S. H., Switching Circuits and Logical Design, John Wiley and Sons, Inc., New York, N. Y., pp. 453-659, 1958.

6. Mealy, G. H., "A Method for Synthesizing Sequential Circuits," Bell System Technical Journal, Vol. 34, pp. 1045-1079, September 1955.

7. Aufenkamp, D. D., and Hohn, F. E.,"Analysis of Sequential Machines," Institute of Radio Engineers Transactions on Electronic Computers, Vol. E C-6, pp. 276-285, December 1957.

8. Aufenkamp, D. D., "Analysis of Sequential Machines II," IRE Trans. on Electronic Computers, Vol. E C-7, pp. 299-306, December 1958.

9. Ginsburg, S., "A Synthesis Technique for Minimal State Machines," IRE Trans. on Electronic Computers, Vol. E C-8, pp. 12-24, March 1959.

10. Ginsburg, S., "On the Reduction of Superfluous States in a Sequential Machine," Journal of the Association of Computing Machines, Vol. 6, pp. 259-282, April 1959.

11. Ginsburg, S., "A Technique for the Reduction of a Given Machine to a Minimal-State Machine," IRE Trans. on Electronic Computers, Vol. E C-8, pp. 346-355, September 1959.

12. Ginsburg, S., "Synthesis of Minimal-State Machines, "IRE Trans. on Electronic Computers, Vol. E c-8, pp. 441-449, December 1959.

13. Paull, M. C., and Mager, S. E., "Minimizing the Number of States in Incompletely Specified Sequential Switching Functions," IRE Trans. on Electronic Computers, Vol. EC-8, pp. 356-367, September 1959.

14. Marcus, M. P., Switching Circuits for Engineers, Prentice-Hall, Englewood Cliffs, N. J., pp. 184-187, p. 189, 1962.

15. Burroughs Algebraic Compiler, A Reference Manual. Bulletin 220-21011-D, Burroughs Corporation, Detroit, Michigan, January 1961.

16. Armstrong, D. B., "A Programmed Algorithm for Assigning Internal Codes to Sequential Machines," IRE Trans. on Electronic Computers, Vol, EC-11, pp. 466-472, August 1962.